AD-A093 084 YALE UNIV NEW HAVEN CT DEPT OF COMPUTER SCIENCE F/G 9/2

RETRIEVAL AND ORGANIZATIONAL STRATEGIES IN CONCEPTUAL MEMORY: A--ETC(U)

NOV 80 J L KOLODNER N00014-75-C-1111

UNCLASSIFIED RR-187 NL

1 of 3

AD A093084

Retrieval and Organizational Strategies
in Conceptual Memory:
A Computer Model

November 1980

Research Report #187

Janet L. Kolodner

YALE UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

This work was presented to the Graduate School of Yale University in candidacy for the degree of Doctor of Philosophy. The author is presently a member of the Department of Computer Science Georgia Institute of Technology.

Retrieval and Organizational Strategies
in Conceptual Memory:
A Computer Model

November 1980

Research Report #187

Janet L. Kolodner

This work was supported in part by the Advanced Research Projects Agency of the Department of Defense and monitored under the Office of Naval Research under contract N00014-75-C-1111, and in Part by the National Science Foundation under contract I ST 7918463.

9) Research rept.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 14) RR-187 #187	2. GOVT ACCESSION NO. AD-A093	3. RECIPIENT'S CATALOG NUMBER 084
4. TITLE (and Subtitle) 6) Retrieval and Organizational Strategies in Conceptual Memory: A Computer Model.		5. TYPE OF REPORT & PERIOD COVERED Technical
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) 10) Janet L. Kolodner		8. CONTRACT OR GRANT NUMBER(s) 15) N00014-75-C-1111, NSF-IST79-18463
9. PERFORMING ORGANIZATION NAME AND ADDRESS Yale University-Computer Science Department 10 Hillhouse Avenue New Haven, Connecticut 06520		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Advanced Research Projects Agency 1400 Wilson Boulevard Arlington, Virginia 22209	12	12. REPORT DATE 11 November 1980
		13. NUMBER OF PAGES 247
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Office of Naval Research Information Systems Program Arlington, Virginia 22217		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution of this report is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Artificial Intelligence
Natural Language Processing
Term Memory
Episodic Memory
Intelligent Information Retrieval

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

People effortlessly recall past events and episodes in their lives many times in the course of a normal day. A reasonable goal in the design of computer programs is to construct a memory with that same capability. To facilitate human-like retrieval of events from a computer memory, we must first specify a reasonable memory organization. We must then design updating and retrieval processes to build up and access that information. This thesis will present such a theory, and will describe a computer program called

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

407 [illegible]

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

CYRUS which implements that theory.

CYRUS (Computerized Yale Retrieval and Updating System) stores and retrieves episodes in the lives of Secretaries of State Cyrus Vance and Edmund Muskie. When new events are added to its memory, CYRUS integrates them into memory along with the events it already knows about. CYRUS can then answer questions posed to it in English about the events it stores.

The algorithms and memory organization used in CYRUS have been developed by examining the way people answer questions requiring extensive memory search. Its reconstructive processes include instantiation strategies, which construct and elaborate on contexts for search, and search strategies, which direct construction.

Reconstructive processes require a vast store of generalized knowledge in order to be applied. Reconstructive retrieval implies a memory organization which organizes both generalized information about different types of events and distinguishing features of particular events. CYRUS' memory is self-organizing. When given a new fact about Vance or Muskie, it integrates the new event into its already-existing memory organization. In the process, it updates its generalized information and indexes the new event in the appropriate places. CYRUS can be seen as both a model of human memory and an intelligent information retrieval system.

-- OFFICIAL DISTIRUBTION LIST --

Defense Documentation Center Cameron Station Alexandria, Virginia 22314	12 copies
Office of Naval Research Information Systems Program Code 437 Arlington, Virginia 22217	2 copies
Dr. Judith Daly Advanced Research Projects Agency Cybernetics Technology Office 1400 Wilson Boulevard Arlington, Virginia 22209	3 copies
Office of Naval Research Branch Office - Boston 495 Summer Street Boston, Massachusetts 02210	1 copy
Office of Naval Research Branch Office - Chicago 536 South Clark Street Chicago, Illinois 60615	1 copy
Office of Naval Research Branch Office - Pasadena 1030 East Green Street Pasadena, California 91106	1 copy
Mr. Steven Wong New York Area Office 715 Broadway - 5th Floor New York, New York 10003	1 copy
Naval Research Laboratory Technical Information Division Code 2627 Washington, D.C. 20375	6 copies
Dr. A.L. Slafkosky Commandant of the Marine Corps Code RD-1 Washington, D.C. 20380	1 copy
Office of Naval Research Code 455 Arlington, Virginia 22217	1 copy

Office of Naval Research Code 458 Arlington, Virginia 22217	1 copy
Naval Electronics Laboratory Center Advanced Software Technology Division Code 5200 San Diego, California 92152	1 copy
Mr. E.H. Gleissner Naval Ship Research and Development Computation and Mathematics Department Bethesda, Maryland 20084	1 copy
Captain Grace M. Hopper NAICOM/MIS Planning Board Office of the Chief of Naval Operations Washington, D.C. 20350	1 copy
Dr. Robert Engelmore Advanced Research Project Agency Information Processing Techniques 1400 Wilson Boulevard Arlington, Virginia 22209	2 copies
Professor Omar Wing Columbia University in the City of New York Department of Electrical Engineering and Computer Science New York, New York 10027	1 copy
Office of Naval Research Assistant Chief for Technology Code 200 Arlington, Virginia 22217	1 copy
Major J.P. Pennell Headquarters, Marine Corp. (Attn: Code CCA-40) Washington, D.C. 20380	1 copy
Computer Systems Management, Inc. 1300 Wilson Boulevard, Suite 102 Arlington, Virginia 22209	5 copies
Ms. Robin Dillard Naval Ocean Systems Center C2 Information Processing Branch (Code 8242) 271 Catalina Boulevard San Diego, California 92152	1 copy

ABSTRACT

Retrieval and Organizational Strategies
in Conceptual Memory:
A Computer Model

Janet Lynne Kolodner

People effortlessly recall past events and episodes in their lives many times in the course of a normal day. A reasonable goal in the design of computer programs is to construct a memory with that same capability. To facilitate human-like retrieval of events from a computer memory, we must first specify a reasonable memory organization. We must then design updating and retrieval processes to build up and access that information. This thesis will present such a theory, and will describe a computer program called CYRUS which implements that theory.

CYRUS (Computerized Yale Retrieval and Updating System) stores and retrieves episodes in the lives of Secretaries of State Cyrus Vance and Edmund Muskie. When new events are added to its memory, CYRUS integrates them into memory along with the events it already knows about. CYRUS can then answer questions posed to it in English about the events it stores.

The algorithms and memory organization used in CYRUS have been developed by examining the way people answer questions requiring extensive memory search. Its reconstructive processes include <u>instantiation strategies</u>, which construct and elaborate on contexts for search, and <u>search strategies</u>, which direct construction.

Reconstructive processes require a vast store of generalized knowledge in order to be applied. Reconstructive retrieval implies a memory organization which organizes both generalized information about different types of events and distinguishing features of particular events. CYRUS' memory is self-organizing. When given a new fact about Vance or Muskie, it integrates the new event into its already-existing memory organization. In the process, it updates its generalized information and indexes the new event in the appropriate places. CYRUS can be seen as both a model of human memory and an intelligent information retrieval system.

Preface

Someday we expect that computers will be able to keep us informed about the news interactively. People have imagined being able to ask their home computers questions such as "What's going on in the world?", "How long has the war in the Persian Gulf been going on?", or "Where is President Carter today?" In order to answer questions such as this, computers must have the capability of remembering events. People remember past episodes in their lives many times in the course of a normal day, but they are largely unaware of the processes they use to remember.

Remembering involves finding requested information in memory. Thus, remembering requires processes both for memory search and for organizing information in memory so that it can easily be found. This thesis will present a theory of remembering and a computer program that answers questions about Cyrus Vance and Edmund Muskie. Because people are good at remembering, the theory will draw on observations of people, and the computer program will model the processes people seem to use.

This thesis has two purposes, then. It presents a theory of remembering which can be used in implementing intelligent computer systems. The theory also attempts to explain how people's memories work, and makes predictions about the organization of human memory.

Acknowledgments

There are numerous people I would like to thank who helped make this thesis possible.

First, I would like to thank my thesis advisor, Professor Roger Schank. In the four years I have been at Yale, he's taught me everything I know about doing research and writing a thesis. He has been a source of inspiration to me, and I know his affect on me will last a long time. No one can convince me that there's a better advisor around.

I'd also like to thank Professor Drew McDermott and Dr. Chris Riesbeck, who both read my thesis and made invaluable comments concerning its content and style. Drew and Chris were also a great help in solving my programming problems. They, along with Gene Charniak, taught me what I know about AI programming.

Professors Bob Abelson and John Black read my thesis, and pointed me to the relevant psychological literature. I've enjoyed my discussions with them about the relevance of my research to psychology. Michael Dyer's and Dr. Ray Gibbs' comments on different chapters of the thesis were a great help. Ann Drinan gave me advise about my writing style. Professor Wendy Lehnert also read my thesis. If the thesis is still unreadable, it's because I didn't always take everyone's advice. Thank you to Drew, Bob, and Wendy for serving on my committee.

Discussions with Mike Lebowitz about memory organization, thesis writing, and life in general, were particularly useful and enjoyable. So were discussions with Larry Birnbaum. Thank you.

A number of people in the AI project have worked on CYRUS, adding their own modules to the system or adding to my implementation. Martin Korsin wrote the question parser that CYRUS uses, making it possible for CYRUS to communicate with people who can't type CD. Larry Birnbaum acted as a consultant to Marty on parsing. Rod McGuire wrote CYRUS' English language generator. CJ Yang, Alan Cypher, Tim Scerba, Warren Odom, Natalie Dehn, and Steve Lytinen also contributed to CYRUS' implementation.

The interface between CYRUS and FRUMP has taken the time of innumerable people. Martin Korsin, Nat Mishkin, and Michael Dyer worked at different times on the interface. Nat has also provided valuable computer expertise. John McCreery, Anne Hafer, Kris Hammond, Lewis

Johnson, and Jim Hendler all contributed to extending FRUMP for the domains CYRUS required. Thanks also to Jerry DeJong who acted as the FRUMP consultant for those people.

Thank you also to the rest of the people who have been part of the Yale AI project in the four years I was there. They have helped to made it an exciting and enjoyable place to do research. Among those people are Mark Burstein, Professor Jaime Carbonell, Gregg Collins, Professor Richard Cullingford, Glenn Edelson, Bill Ferguson, Margot Flowers, Linda Fusco, Dr. Anatole Gershman, Dr. Jim Galambos, Professor Richard Granger, Dr. Shoshana Hardt, Pete Johnson, Patti Oronzo, Karen Parady, Dr. Jerry Samet, Professor Mallory Selfridge, Dr. Stephen Shwartz, Stephen Slade, Walter Stutzman, Mike Ward, and Professor Robert Wilensky.

There are a number of people from outside of Yale and AI who nevertheless had a great deal to do with my getting a Ph. D. Professor Jacques Cohen, and the staff at Brandeis University's Feldberg Computer Center when I was an undergraduate there, made computer science seem exciting to me. My introduction to AI came from their stories about what computers would be like in the future.

Michael Klein, my husband, has provided considerable moral support. He has patiently put up with my doubts and frustrations, saying more times than either of us can count "Don't worry Janet, it will be alright". His more passive contribution to the thesis was the fact that he was living in New Haven when I was applying to graduate school.

Lastly, I'd like to thank my parents, Meyer and Sarah Kolodner, who instilled in me and encouraged my joy of learning, curiosity, independence, and perseverence. Without those qualities, I would never have considered going to graduate school, and writing a thesis would have been impossible.

Table of Contents

CHAPTER 1

The Nature of a Long Term Memory for Events

1.1 Introduction

In the course of a normal day, people easily remember past experiences. In addition, they have new experiences, and are later able to recall those. Consider, for example, the following question:

(Q1-1) Mr. Vance, when was the last time you saw an oil field in the Middle East?

Answering this question requires retrieval of an event from memory. This thesis will address the problem of retrieving events from a long term memory. There are three major questions that must be answered in addressing this problem:

1. What are the processes for retrieving events from memory?

2. What memory organization do the retrieval processes imply?

3. What are the processes for adding new events to memory, and how does memory organization change as new events are added?

These questions cannot be answered independently of each other. The organization of memory constrains the types of retrieval and updating processes the memory can have. On the other hand, memory organization, and therefore procedures for adding information to memory, must be designed based on retrieval requirements. In considering these problems, I will present a computer implementation of a system that retrieves events from memory and automatically reorganizes itself as new events are added to it.

Like people, a computer system should be efficient at retrieving events. Since people are the only available model we have of intelligent fact retrieval, and since their algorithm might be the only

efficient algorithm for retrieval, it makes sense to attack these problems by modeling the way people remember.

The next two sections of this chapter will present an overview of some of the problems involved in retrieving events from memory and some of the requirements that the retrieval process places on memory's organization. The computer program will then be presented.

1.2 Retrieving events from a long term memory

Consider, again, question (Q1-1), repeated below:

(Q1-2) Mr. Vance, when was the last time you saw an oil field in the Middle East?

If "seeing oil fields" were one of memory's categories, then this question would be fairly easy to answer. "Seeing oil fields" would be selected for search. If it indexed an episode in the Middle East, that episode could be retrieved from it. Similarly, if "seeing objects" were a memory category, it could be selected for retrieval and events in the Middle East and events at oil fields could be retrieved.

What if neither of these categories exist? We can imagine the following reasoning process to answer the question:

A1: An oil field is a large sight, perhaps I saw an oil field during a sightseeing episode in the Middle East.

Using information about episodic contexts associated with "large sights", a "sightseeing" category can be chosen for retrieval. If there is a sightseeing category in memory, then it will be selected. Its contents can be searched for an episode at oil fields in the Middle East. If the sightseeing category organized its episodes according to the type of sight and its part of the world, and if there had been an episode in the Middle East at an oil field, then "a sightseeing episode at an oil field in the Middle East" could be retrieved.

But what if sightseeing episodes were not organized in a category according to the type of sight or by their place in the world? In that case, the following elaboration of the initial retrieval specification might be appropriate to answer the question:

A2: Which countries in the Middle East have oil fields? Iran and Iraq have oil fields, and Saudi Arabia does. ...

If sightseeing episodes are organized according to the country they took place in, then elaborating on "the Middle East" and specifying particular countries in the Middle East would enable retrieval of episodes that took place in each of those places. Instead of searching for "sightseeing at an oil field in the Middle East", search for each of

the more specific episodes "sightseeing at an oil field in Iran", "sightseeing at an oil field in Iraq", etc. could be attempted.

Retrieving sightseeing experiences directly is not the only way to find an experience of seeing oil fields in the Middle East. Retrieval can proceed by searching for an episode that might refer to the target episode. Since sightseeing in the Middle East would have had to happen during a trip to the Middle East, retrieving a trip to the Middle East could aid retrieval of an appropriate sightseeing experience. Thus, the following reasoning would also be appropriate to answer the question above.

> A3: In order to go sightseeing in the Middle East, I would have had to have been on a trip there. On a vacation trip, I wouldn't go to see oil fields, so I must have been taken to oil fields during a diplomatic trip to the Middle East. Which countries might have taken me to see their oil fields? Saudi Arabia has the largest fields, perhaps they took me to see them. ...

Why does it seem reasonable to search for "trips" when a "sightseeing" episode should be retrieved? How can search for alternate events be constrained? Only alternate contexts that might be related to an event targeted for retrieval should be searched for.

In general, for search to be constrained to relevant contexts, event categories must hold generalized information concerning the relationships of their events to other types of episodes. Generalized information associated with a memory category should be information common to individual items in the category, i.e., their similarities.

This retrieval process can be seen as a process of reconstructing what might have happened and checking memory to make sure it did. To retrieve an episode of "seeing oilfields", a hypothesis was made about the type of event it might have been (sightseeing), where it might have happened (Iran, Iraq, Saudi Arabia, etc.), and what else might have been going on at the time (a trip).

Judging from this example, the process of retrieval requires at least the following processes:

1. selection of a category for search

2. search within the category for the targeted event

3. elaboration on the specification of the event to be retrieved

4. search for episodes related to the target event

We call the processes that do these tasks retrieval strategies. Retrieval strategies are the rules that direct memory search. There are two kinds of retrieval strategies -- strategies that guide search for a

requested event, and processes that select a category to retrieve it from and elaborate on its features. The second set of strategies can be thought of as strategies for constructing search keys. The processes which direct search are called _search strategies_, while those which construct and elaborate on search keys are called _instantiation strategies_.

When a particular category is chosen for search by a search strategy, a specification of what to look for in the category (i.e., a search key) must be constructed by an instantiation strategy. Once it has been decided to search the "sightseeing" category to answer question (Q1-2) above, for example, the "sightseeing" episode must be specified further as "sightseeing at oil fields in the Middle East". Chapters 2, 3, and 4 will present retrieval strategies in detail.

1.3 Requirements on the memory organization

There are a number of features of event memory which we can see in people which are also desirable in a computer system designed to retrieve events. People learn new things every day, but they do not get slower at remembering as new events are added to their memories (Smith, et al., 1978).* In a computer system, too, _retrieval should not slow down significantly as new events are added to memory_. This requirement constrains both the retrieval processes and the memory organization. In terms of the retrieval processes, it requires the following:

> Retrieval from an event category must be able to happen without enumeration of the items in the category.

Otherwise, the retrieval process would slow down with each new addition to a category. In addition, it is clear that people are not able to enumerate categories, as evidenced by the difficulties they have in listing similar experiences they have had. People, for example, find it difficult to list off all the the museum experiences they've had (Schank and Kolodner, 1979).

The traditional solution within computer science to the efficiency problem just mentioned is to index items within categories. An event should be indexed in a category by those of its features that are salient to the category. In that way, specification of an indexed

---

* In fact, there is some debate about this among psychologists. Anderson (1974, 1976) cites the "fan effect" as evidence that retrieval slows down with the addition of new items about a particular concept. Smith (1978), on the other hand, has shown that retrieval does not slow down with the addition of new items when context is guiding the retrieval. Reder and Anderson (1980) later conclude that when people make consistency judgements, rather than retrieving actual facts from memory, there is no interference to slow down the retrieval process.

feature will enable retrieval of items with that feature without enumerating the whole category. As was illustrated in the past example, however, the features specified in a retrieval request might not correspond to those that are indexed. In that case, additional related indexed features must be inferred. We shall see that the generalized information associated with memory categories is used to make those inferences.

If memory categories are heavily indexed by salient features, retrieval processes will have a large selection of features to specify, any of which might specify a target event. The retrieval process will be made easier since the easiest elaborations can be attempted first.

The richer the indexing, however, the more space is needed for storage. <u>Indexing must be controlled so that memory does not grow exponentially</u>. Similarities between events can be used to control indexing. If memory keeps track of the similarities between events within a category, then indexing can be limited to the differences between events. Thus, if almost all the events in a "diplomatic meetings" category are with foreign diplomats, indexing them according to the occupations of their participants would be redundant and therefore unnecessary. If, however, one of those meetings were with someone other than a foreign diplomat, indexing the meeting by that feature would differentiate it from other events in the category.

Similarities which constrain indexing correspond to the generalized information necessary for retrieval. Chapters 6 and 7 will discuss integration of new events into memory, and will present a generalization process as a control on indexing.

Finally, <u>a memory for events should maintain itself</u>. We have stated that retrieval of events from memory requires use of generalized information, that events are organized in event categories which also have generalized information associated with them, and that retrieval need not slow down as new events are added to memory. Human memories are constantly being updated. If retrieval does not slow down as new events are added to memory, then memory must be able to maintain its organization, creating new conceptual categories when necessary and building up required generalized information. Maintainance of memory organization will be described in chapter 7.

1.4 CYRUS

CYRUS (Computerized Yale Retrieval and Updating System) (Kolodner, 1978, Schank and Kolodner, 1979) is a computer program which implements the theory of long term memory presented in this thesis. CYRUS is meant to be both an intelligent information retrieval or data base system, and a model of the way human long term memory for events works. CYRUS has been designed to keep track of events in the lives of important people. It answers questions posed to it in English pertaining to that information. Thus, it both organizes events in its memory and searches memory reconstructively to retrieve them.

Currently CYRUS can be used with two data bases -- one holds the experiences of former Secretary of State Cyrus Vance, and one holds those of the present Secretary of State, Edmund Muskie. Cyrus Vance was initially chosen as the subject for the memory for two reasons -- (1) he is in the news often enough to make a self-updating data base interesting, and (2) few enough of the types of activities he does are reported to make initial organization of the system's memory feasible. When Vance resigned from office, a separate data base for Muskie, who took over for Vance, was started. No changes had to be made to the program.

Events in CYRUS' memory are organized in event categories according to their differences. These categories, called E-MOPs (Episodic Memory Organization Packets), will be explained in the next section of this chapter. CYRUS' E-MOPs include "diplomatic trips", "diplomatic meetings", "negotiations", "speeches", and "state dinners" — the normal activities of a diplomat.

When CYRUS adds new events to its memory, it automatically organizes them there so that they can be easily retrieved using the constructive memory techniques of context construction and elaboration. As it adds a new event to memory, CYRUS indexes the event in the appropriate categories, reorganizing those categories as necessary, and creating category/sub-category hierarchies specific to the experiences it is given. Although the Vance and Muskie data bases started out with the same categories, because of the differing experiences of the two men, the organization of the two memories after events were added is quite different.

Because CYRUS does not have the capability to understand stories expressed in Natural Language, stories must be read and conceptual representations for them produced before CYRUS can add them to its memory. The information in CYRUS' memory has been collected from news stories in two different ways. Most of the representations for events concerning Vance were hand-coded from newspaper stories. The Muskie data base, on the other hand, has had all of its information collected by FRUMP (DeJong, 1979), a computer program that reads stories from the UPI wire and produces summaries. When FRUMP produces a summary of a story about Muskie or Vance, it sends the conceptual representation underlying the summary to CYRUS. CYRUS then adds the new events to its memory and can later be queried about them.

CYRUS answers questions posed to it in English, and gives English answers. The following are actual QA dialogs with CYRUS. The first is a dialog with the Vance memory, the second with the Muskie data base. The remainder of this thesis will explain the processes and organization of a memory with CYRUS' capabilities.

```
***************************

@CYRUS

ARE YOU INTERESTED IN MUSKIE OR VANCE?  (M OR V) : *VANCE


*(DIALOG2)

>When was the last time you were in Egypt?

   inferring a diplomatic trip
   answering question using time context

 ON DEC 10 1978.

>Why did you go there?

   answering question using previous context

 TO NEGOTIATE THE CAMP DAVID ACCORDS.

>Who did you talk to there?

   inferring undifferentiated political meeting
   answering question using previous context

 WITH ANWAR SADAT.

>Has your wife ever met Mrs Begin?

   inferring a social political occasion

 YES, MOST RECENTLY AT A STATE DINNER IN ISRAEL IN JAN 1980.

>What heads of state have you met?

   inferring a diplomatic meeting
  searching directly for $MEET
    ...
  applying strategies to search memory
    ...
     searching for sM-CONFERENCE
      ...
     searching for sM-VIPVISIT
      searching sM-VIPVISIT instance for $MEET
      ...
     searching for I-NEGOTIATE
      ...

 KING HASSAN, MENACHIM BEGIN, MUSTAFA KHALIL, IAN SMITH,
 PRESIDENT ASSAD, KING HUSSEIN, AND ANWAR SADAT.
```

```
>Last time you went to Saudi Arabia, where did you stay?

   inferring a diplomatic trip
   answering question using time context

 IN A PALACE IN SAUDI ARABIA ON SEP 23 1978.

>Did you go sightseeing there?

    answering question using previous context

 YES, AT AN OILFIELD IN DHARAN ON SEP 23 1978.

***************************

@CYRUS

ARE YOU INTERESTED IN MUSKIE OR VANCE?  (M OR V) : *MUSKIE


*(DIALOG2)

>Where are you today?

  inferring a diplomatic trip

 PROBABLY IN THE UNITED STATES ON JUN 14 1980.

>Where were you three weeks ago?

  inferring a diplomatic trip

 IN EUROPE.

>Why did you go?

   answering question using previous context:

 TO TALK TO ANDREI GROMYKO.

>Who did you talk to?

   inferring undifferentiated political meeting
   answering question using previous context

 TO NATO IN BRUSSELS ON MAY 14 AND TO ANDREI GROMYKO IN
 VIENNA.

>Did you give a speech?

   answering question using previous context

 YES, ONE TO A GROUP OF REPORTERS AND ONE TO NATO.
```

>Are you going to Asia?

inferring a diplomatic trip

YES, THIS MONTH.

>Who will you talk to?

inferring undifferentiated political meeting
answering question using previous context

TO NATO IN ANKARA, TURKEY.

****

1.5 An overview of the retrieval process

Although retrieval strategies are the primary part of the entire retrieval process, they are only one part of it. Retrieval requires understanding a statement or question, extracting the part of it that should be searched for, and after retrieving it, deciding which parts of the event retrieved should be expressed. The retrieval process which will be assumed is based on Lehnert's (1978) description of the question-answering process. In order to answer a question, the question must be analyzed and its conceptual category and question concept must be extracted. The _question concept_ (Lehnert, 1978) of a question is the part of the question that must be searched for, i.e., its _target event_. The target event is searched for using the retrieval processes which have been introduced and which will be described in more detail in later chapters. When an event corresponding to the target event is found in memory, processes associated with the question category are applied to it to formulate an answer. If the question asks "why", for example, the causes of the retrieved event are extracted and expressed. The entire retrieval process can be described by the following algorithm:

The Retrieval Loop

```
          +--------------------+
          | 1. get a question  |  <------+
          +--------------------+         |
                   \ /                   |
            +----------------+           |
            | 2.  parse it   |           |
            +----------------+           |
                   \ /                   |
     +--------------------------+        |
     | 3.  resolve references   |        |
     +--------------------------+        |
                   \ /                   |
  +----------------------------+         |
  | 4.  extract question type  |         |
  |      and target concept    |         |
  +----------------------------+         |
                   \ /                   |
          +--------------------+         |
          | 5.  search memory  |         |
          +--------------------+         |
                   \ /                   |
        +----------------------+         |
        | 6.  formulate answer |         |
        +----------------------+         |
                   \ /                   |
        +----------------------+         |
        | 7.  generate answer  | --------+
        +----------------------+
```

Figure 1-1

The retrieval processes and strategies which will be presented are the processes used to search memory (step 5 above). When the term "retrieval" is used throughout this thesis, it will refer to memory search, the primary step in the retrieval process. As we shall see in chapter 8, some of the processes for memory search can also be used to extract an answer to a question (step 6) once an event has been retrieved from memory.

The search process itself can be seen as *a process of specifying and elaborating on contexts for search*. To search memory for a particular targeted event, it is necessary to first specify a memory category that event might be found in. To search for the targeted event within a category, it is necessary to specify the types of event features that organize the category, i.e., the types of features the category uses as indices. This often requires *elaboration* of the information given in the target event. If an event targeted for retrieval cannot be retrieved from the selected category, it is often

useful to search for an alternate related context that might refer to a target event. The following diagram summarizes that process:

Memory Search

```
                   +---------------------+
                   | choose a context    | <-------------+
                   |  for search         |               |
                   +---------------------+               |
                     /             \                     |
                 fail               succeed              |
                 /                    \                  |
            retrieval     +------------------------+     |
            failure       | choose indices for the |     |
                          | targeted event in the  |     |
                          |       category         |     |
                          +------------------------+     |
                              | succeed        \         |
                             \ /                fail     |
                   +------------------------+     \      |
       +-------->  | follow the indices     |      |     |
       |           +------------------------+      |     |
       |                      |                    |     |
       |                     \ /                   |     |
       |           +------------------------+      |     |
       |           |  found an event ?      |      |     |
       |           +------------------------+      |     |
       |                  /          \             |     |
       |                 no          yes           |     |
       |                /              \           |     |
       |   +-------------------+    return the     |     |
       |   |  elaborate        |      event        |     |
       |   +-------------------+                  /      |
       |          /        \                     /       |
       |      succeed      fail                 /        |
       +----------+          \                 /         |
                         +---------------/--+            |
                         | alternate context |           |
                         |  search           |           |
                         +-------------------+           |
                            /         \                  |
                       succeed        \ fail             |
                          /            \-----------------+
                    return events
```

Figure 1-2

Retrieval from memory will be described in more detail in section 1 of the thesis. In order to explain the retrieval process, it is first necessary to give an overview of the memory organization. The next

section will do that.

1.6 An overview of the memory organization

The memory organization implied by the retrieval process proposed above is based on conceptual categories for events. As stated in that chapter, these categories index similar episodes according to their differences and keep track of the similarities between the events. Individual episodes in a category cannot be enumerated, however. These categories will be referred to as Episodic Memory Organization Packets (E-MOPs), or generically as "MOPs". These structures are related to Schank's (1980) MOPs and to Lebowitz's (1980) S-MOPs, but the concerns in defining MOPs and S-MOPs were different than those in defining E-MOPs.*

1.6.1 What kinds of situations correspond to E-MOPs?

E-MOPs should both organize events for retrievability and organize generalized information about the events. When situations are too much alike or too different, they will not correspond to useful E-MOPs. Events are organized in E-MOPs according to their differences from its norms. Thus, E-MOPs should correspond to situations which are similar to each other, but which can be distinguished by domain-relevant criteria. The situations should be similar to each other so that generalized information necessary for retrieval and later indexing can be maintained on the E-MOP. Since retrieval strategies can predict only domain-related information, their differences should be domain-relevant so that retrieval strategies can be used to retrieve them. Domain-relevant features of meetings, for example, include their topics, participants, and purpose. For homebuying, domain-relevant features are the condition and price of the house.

----

* In particular, Schank's (1980) concern was with showing the inter-relatedness of structures in memory. Thus, in his example domain of professional office visits, he described how visits to doctors, dentists, lawyers, and other professionals are similar, and how the structures they are stored in are related. My concern, on the other hand, is with the processes for retrieval of individual episodes, and the organizational requirements those processes place on memory. If my domain were professional office visits, I would be describing how particular visits are stored in memory in relation to each other and the generalized information that would allow them to be retrieved.

Lebowitz (1980) used S-MOPs to store terrorism events derived from newspaper stories. E-MOPs are more experientially oriented and hold more detailed information than his S-MOPs.

"Speeches", for example, would be a good E-MOP for a person who does not go to or give too many speeches because its events have a similar structure, but can be differentiated. They consist of the speechmaker being introduced, giving his speech, and receiving questions from the audience. Second, they normally take place in large lecture halls. Two aspects of speeches which can differentiate them are the speaker and the topic of the speech.

On the other hand, if a person gave a number of speeches every day, all about the same topic, then "speeches" would not be a good E-MOP for that person. The speeches would not be easily distinguishable from each other. Although "speeches" would organize a great deal of generalized information about speeches, it could not provide discriminability for the events. We shall see that when an E-MOP is overwhelmed by similar episodes in that way, retrieval strategies can use the generalized information associated with the E-MOP to search for related situations.

"Meeting somebody" or "talking to somebody", also are not usually good E-MOPs. They happen frequently and in many different situations. Thus, there is not a great deal of generalized information which can be built up about them, and there are no domain-related criteria for distinguishing them. They are best stored as part of the situation in which they occur. If one talked to somebody or met somebody during a conference, it would best be stored as part of the conference situation.

In general, situations which are too much alike are not good E-MOPs because their instances cannot be distinguished. Those which are too different are not good because no generalized information can be built up about them. We will see in later chapters that generalized information is necessary to direct retrieval and constrain indexing.

1.6.2 The internal organization of an E-MOP

The internal organization of an E-MOP can be described as a net in which each node in the net is either an E-MOP or an event. Each E-MOP has two important aspects -- (1) generalized information characterizing its episodes, and (2) tree-like structures that index those episodes by their differences. As for Schank's MOPs, the generalized information associated with an E-MOP is called its <u>content frame</u>. An E-MOP's content frame holds information describing its events, such as their usual participants, locations, and topics, and their usual relationships to other events.

One of the E-MOPs CYRUS uses is "diplomatic meetings". Each of Vance's or Muskie's diplomatic meetings that are entered into CYRUS' memory are indexed in that E-MOP. CYRUS knows that the participants of diplomatic meetings are foreign diplomats, that their topics are international contracts, that they include discussion between the participants about the topic, that their goal is usually to resolve a disputed contract, and that they are normally part of negotiations. That is some of the information that makes up the content frame of "diplomatic meetings". In addition, CYRUS' E-MOP for Vance's diplomatic meetings holds the information that Vance is the actor in those

meetings. That content frame is illustrated below.

"diplomatic meetings" content frame

```
content frame:  the actor is Cyrus Vance
                participants are foreign diplomats
                topics are international contracts
                participants talked to each other
                goal is to resolve disputed contract
                instrumental to negotiations
```

Figure 1-3

The second important feature of an E-MOP is its indices. An E-MOP's indices can index either individual episodes or specialized E-MOPs. When an E-MOP holds only one episode with a particular index, that index will point to the individual episode. When two or more episodes in an E-MOP share the same feature, its corresponding index will point to a specialized sub-MOP (with the structure just described) which organizes the subset of events with that feature. In this way, MOP/sub-MOP hierarchies are formed.

Consider, for example, how the following two events are indexed in CYRUS' "diplomatic meetings" MOP.

```
EV1:  $MEET actor (Vance)
            others (Begin)
            topic (the Camp David Accords)

EV2:  $MEET actor (Vance)
            others (Gromyko)
            topic (SALT)
```

Both of these meetings are diplomatic meetings with foreign diplomats about international contracts. One is with Begin about the Camp David Accords (EV1), and one is with Gromyko about SALT (EV2). These two meetings are discriminated in CYRUS' "diplomatic meetings" MOP ($MEET) as follows:

```
                    "diplomatic meetings" — $MEET

content frame:  the actor is Cyrus Vance
                participants are foreign diplomats
                topics are international contracts
                participants talked to each other
                goal was to resolve disputed contract
diffs:              /                     \
              participants                topic
                  / \                      / \
            Begin    Gromyko        SALT    Camp David
              |         |             |       Accords
             EV1       EV2           EV2         |
                                                EV1
```

Figure 1-4

As additional meetings are added to memory, generalized content frame information is refined, and additional indices for events are created. As that is happening, new E-MOPs are created where multiple episodes are indexed. Each of those new E-MOPs has associated with it a content frame based on the similarities between the episodes it indexes. Episodes are indexed in each new E-MOP according to their differences from its content frame. These newly-created specialized E-MOPs inherit content frame properties from the more general E-MOPs they are specialized from, and in addition have their own more specialized content frame information. Thus, E-MOPs and their specializations form a hierarchy discriminated by differences from content frame features.

After many meetings with Begin are added to the memory structure in figure (1-4) above, its organization would include the following:

```
                    "diplomatic meetings"

              (1)  (MOP1)
content frame:  the actor is Cyrus Vance
                participants are foreign diplomats
                topics are international contracts
                participants talked to each other
                goal was to resolve disputed contract
diffs:         /                         \
        participants                   topic
      ../  |    \                   .../  |  \..
    ./  (2)|     \                 /      |    \  (3)
   /    Dayan  Gromyko          SALT      |     Jerusalem
   |       |     |              |         |         |
   |      EV4   EV2            EV2        |        EV3
(4)|                                 (5)  | (MOP3)
Begin  (MOP2)                     Camp David Accords
 content frame:                      content frame:
   participants include Begin          topic is the CDA
   topic concerns Israel               participants are Israeli
      and Arabs                        specialization of MOP1
   specialization of MOP1          diffs:     |
 diffs:     |                                 | participants
            | topic                  (8)    / \ (9)
 (6)      / \  (7)                     Begin   Dayan
 Jerusalem   Camp David Accords       (MOP4)     |
    |            (MOP4)                         EV4
   EV3
```

Figure 1-5

MOP1, "diplomatic meetings", is a refined version of the diplomatic meetings MOP in figure (1-4), while MOP2 and MOP3 -- "diplomatic meetings with Begin" and "diplomatic meetings about the Camp David Accords", respectively -- are at the points in MOP1 where the meeting with Begin about the Camp David Accords is indexed. SALT and Gromyko remain indices to EV2, an individual event, since no additional similar meetings were added to the MOP. Thus there would be no MOPs created at those index points until additional meetings about SALT or with Gromyko were added. Index points (2) and (3) index meetings with Dayan and about Jerusalem, respectively, in "diplomatic meetings". Index points (6), (7), (8), and (9) are new indices in MOP2 and MOP3, and index differences from the content frames of those newly created MOPs. The meeting with Dayan and the meeting about Jerusalem in MOP1, and also in appropriate specialized E-MOPs.

An organization such as this provides rich cross-indexing of events in memory. Specification of any discriminating set of event features within an E-MOP should allow retrieval of the corresponding event. Using a richly indexed organization such as this, enumeration of a

memory category should never be necessary for retrieval. Rather, specification of indexed features allows search to be directed only to categories and sub-categories whose events are relevant.

1.6.3 Classifying E-MOPs

E-MOPs can be classified according to the kinds of information they hold in their content frames. As we shall see later, this classification can constrain search strategy application, since only strategies appropriate to the kinds of information an E-MOP holds in its content frame need be applied.

Scripts (Schank and Abelson, 1977) are events which take place over a short period of time, usually in one location, and with very stereotyped sequences of events. As such, they define a well-specified event context and are one type of E-MOP. Some of the scripts CYRUS uses are diplomatic meeting ($MEET), speech ($SPEECH), fly ($FLY), and welcoming ceremony ($WELCOME). Scripts are the smallest event unit which can be searched for. Individual actions (such as talking) must be found by searching for episodes (often scripts) they could have been embedded in.

Scriptal events are often embedded in the sequence of events of larger episodes. Some of the larger types of episodes CYRUS uses are diplomatic trip (sM-VIPVISIT) and summit conference (sM-SUMMIT-CONFERENCE). Diplomatic trips and summit conferences are examples of simple MOPs (Schank, 1980). Simple MOPs have less stereotyped sequences of events than scripts do and often include scriptal situations and other simple MOPs in their sequences of events. A diplomatic trip, for example, includes flying ($FLY), being welcomed ($WELCOME), doing a series of diplomatic activities, and flying home ($FLY). Instead of a standard sequence of events, some simple MOPs have a set of scripts which are their standard ways of being done. Travel (sM-TRAVEL), for example, can be done by flying, driving, taking a train, taking a bus, or taking a boat.

The third kind of E-MOP which can be distinguished is an Intentional MOP or I-MOP (Schank, 1980). I-MOPs are episodes that normally take place over a long period of time and have a standard goal. Negotiating (I-NEGOTIATE), for example, takes place over a long period of time and always has the goal of solving the problem being negotiated. It is not usually embedded in any other episode, except perhaps a more encompassing negotiating episode. Vacation (I-TRIP) is also an I-MOP. Its goal is relaxation.

1.7 A guide to the thesis

The rest of this thesis is divided into three major parts. The first section includes chapters 2, 3, and 4, and concerns memory retrieval. In chapter 2, retrieval processes are introduced. Chapters 3 and 4 go into more detail about retrieval strategies, spelling each one out explicitly.

Chapters 5, 6, and 7 comprise the second section of the thesis. It is about memory organization and its maintenance. Chapter 5 draws conclusions from the retrieval strategies about the kinds of knowledge necessary for applying strategies. Chapter 6 explains criteria for judging indices and gives an algorithm for choosing indices for events in event categories. Chapter 7 discusses the process of adding information to memory and building up the generalized information used by retrieval strategies.

The third section is a summary section. Chapter 8 presents CYRUS in detail, describing where the retrieval and organizational strategies presented fit into the implementation. That chapter will also present aspects of CYRUS′ implementation that have not been explained in previous chapters. Chapter 9 presents a literature review, comparing this work to related research in the fields of psychology, artificial intelligence, data base management, and information retrieval.

The conclusion, chapter 10, presents conclusions about memory organization and retrieval processes, from the point of view of intelligent information retrieval, and from a psychological point of view. In that chapter, limitations of this theory and future research will also be examined.

1.7.1 Surviving the reading

Ten chapters about long term memory is a lot to read, even for people with a strong interest in the subject. In the interests of people who do not have time to read a 400 page tome, I have included a summary at the end of each chapter detailing its major points.

The preface to the retrieval section and chapter 2 together provide a background for reading other chapters of the thesis. Chapter 8 explains the implementation of the memory processes and how they are integrated into the computer program, and will be relevant to readers interested in the program itself.

In addition to chapter 2, those readers who are interested in the processes which comprise retrieval will find chapters 3 and 4 most relevant. Chapter 5 explains the knowledge that memory must hold in order for retrieval strategies to be applied, and also may be of interest to those readers. Readers more interested in memory updating processes will find chapters 6 and 7 most relevant.

CHAPTER 2

Retrieving an Event from Memory

2.1 Introduction

Q: Have you been to Saudi Arabia recently?

A: Yes, most recently last month, to discuss the Camp David Accords with King Khalad and Prince Fahd.

Q: Where did you go afterwards?

A: To Syria. I was touring the Middle East talking to each of the Arab leaders about the Accords.

Q: How many times have you been to the Middle East in the past 6 months?

A: I was in Israel and Egypt this past summer on two separate trips, and after the Camp David Summit, I was in the Middle East to talk to Arab leaders about the Camp David Accords.

Taking part in a dialog or discussion often requires retrieval of past events from memory. The answers given in this dialog would have been reasonable ones for Cyrus Vance to give while he was Secretary of State. What are the processes that allow this retrieval to happen? How can a list of events be retrieved from a memory which does not allow enumeration of its categories?

This chapter will explain the retrieval process in detail, focusing on retrieval strategy application. Among the issues which will be addressed are the following:

1. What is the retrieval process?

2. Where do retrieval strategies fit into that process?

3. What kinds of strategies are there?

4. When are they applied?

In order to explain the retrieval strategies, we must first explain the process of searching a particular E-MOP to find an event. Strategies for searching memory derive from the failures in that procedure.

2.2 Retrieving an event from an E-MOP

Events are indexed in memory in E-MOPs according to their differentiating features. An event can be found in an E-MOP, then, by traversing appropriate indices. Consider, again, the E-MOP presented in the preface (repeated below):

```
                       "diplomatic meetings"

              (1)  (MOP1)
content frame:  the actor is Cyrus Vance
                participants are foreign diplomats
                topics are international contracts
                participants talked to each other
                goal was to resolve disputed contract
diffs:         /                      \
        participants                 topic
      ../  |    \                 .../  |  \..
    ./  (2)|     \               /      |    \  (3)
   /    Dayan  Gromyko        SALT      |     Jerusalem
   |      |      |              |       |         |
   |     EV4    EV2            EV2      |        EV3
(4)|                              (5) | (MOP3)
Begin  (MOP2)                  Camp David Accords
 content frame:                   content frame:
   participants include Begin       topic is the CDA
   topic concerns Israel            participants are Israeli
      and Arabs                     specialization of MOP1
   specialization of MOP1        diffs:     |
 diffs:      |                              | participants
             | topic                (8)    / \ (9)
 (6)        / \  (7)                  Begin   Dayan
 Jerusalem   Camp David Accords       (MOP4)    |
    |            (MOP4)                        EV4
   EV3
```

Figure 2-1

To search for "a diplomatic meeting with Gromyko" in this MOP, the index "has Gromyko as a participant" would be followed, and the event indexed at that point (EV2) would be found. Similarly, to find a "diplomatic meeting about SALT", the index "has topic SALT" would be followed and a matching event (EV2) would be found.

Specification of a feature should enable its corresponding index to be followed without needing to enumerate all indices to see which ones match the specification. Thus, we must imagine something like a hash function which allows the indices corresponding to specified features to be followed. There is no need, then, for indices of an E-MOP to be enumerable.

This retrieval process can be characterized as a *traversal* process, a process of following appropriate indices down a tree until an event is found. An event to be retrieved from an E-MOP is called a *target event*, and the features which describe it make up its *context specification*. A target event, or event which must be retrieved, can be said to be *targeted for retrieval*.

As was stated above, rich cross-indexing enables directed search of memory, i.e., search without category enumeration. To ensure directed search, traversal must require, as a first step, specification of paths to traverse, or selection of indices for traversal.

Index selection is based on features specified in the target event. Indices chosen for traversal to find any target event should be features that would have been chosen as indices for that event if it had previously been indexed in the E-MOP. Thus, the same process that is used for index selection when an event is added to memory must be used for index selection during retrieval. When an event is added to memory, the index selection procedure chooses features that should be indexed, and indices are built. During retrieval, the index selection procedure chooses features that would have been indexed if the target event had been added to memory, and those indices are traversed. The actual process of choosing indices will be discussed in chapter 6.

Suppose we wanted to answer the following questions using the MOP illustrated in Figure (2-1) above.

(Q2-1) Have you ever discussed SALT with Gromyko at a diplomatic meeting?

(Q2-2) Have you ever attended a diplomatic meeting about the Camp David Accords with Dayan?

Answering a question requires extraction of its "target concept", that is, the concept that must be searched for in memory, usually the target event. The target concept for (Q2-1) is a "diplomatic meeting about SALT with Gromyko". Answering (Q2-1) requires retrieval of that event. A "diplomatic meeting about SALT with Gromyko" can be retrieved from the structure in Figure (2-1) by traversing either of the indices "has topic SALT" or "has Gromyko as a participant", retrieving the event found at each of those points (EV2), and checking to make sure it has all the required features. Since EV2, found at both index points, is a meeting with Gromyko about SALT, it has all the features of the target event and can be used to answer "yes" to the question.

When a target event specifies an event feature which is unique in an E-MOP, the target event can be found by traversing the index associated with that feature. Question (Q2-1) had two unique features in the "diplomatic meetings" MOP -- its participants and its topic. Either can be followed to retrieve the appropriate event. The target concept for (Q2-2) is a "diplomatic meeting with Dayan about the Camp David Accords". It can be retrieved from figure (2-1) by traversing the index corresponding to the unique MOP feature "has Dayan as a participant" (2). EV4 would be found. It would be checked to make sure it had the topic "the Camp David Accords" (the remainder of the target concept's specification). Since EV4 is a meeting with Dayan about the Camp David Accords, it has all specified features and can be used to answer the question.

There is also a second way the "meeting with Dayan about the Camp David Accords" can be found in that tree. One feature that meeting has is "has topic the Camp David Accords". If that index were traversed,

MOP3, "diplomatic meetings about the Camp David Accords", would be reached. When a specialized E-MOP is reached during the traversal process, its indices are traversed after selecting appropriate indices for traversal to find the target event. In this case, the index "has Dayan as a participant" (9) of MOP3 would be traversed and EV4 would be found, again a sufficient answer to the question.

Because there is no way of knowing before traversal whether or not a feature is unique to an E-MOP, the index associated with each feature selected for retrieval must be traversed. If one is unique, an event will be found and traversal can end. Otherwise, traversal continues at the next E-MOP level. Thus, in answering (Q2-2), both indices "has Dayan as a participant" and "has topic the Camp David Accords" are traversed. Since one is unique, an event is found, and traversal can stop. If, however, the index "has Dayan as a participant" had not been unique (i.e., if there had been more than one meeting with Dayan indexed in the E-MOP, then traversal would have had to continue within the E-MOP at that point, and within the "meetings about the Camp David Accords" MOP.

Thus, traversal is a recursive process involving calculation of differences, or choice of indices, and traversal of those indices. It stops when an event is found, or when there are no more specified indices to be traversed. Thus, if there are multiple paths to a target event, it will be retrieved from the shortest path with all of its indices specified in the target event. We can think of traversal as a breadth-first search which implements parallel traversal of all appropriate indices.

We can thus specify the following algorithm for traversal:

E-MOP Traversal

1. Select possible indices for the target event based on its specified features and their differences from the norm of the E-MOP being traversed.

2. IF there are no indices, THEN return "not found".

3. ELSE follow all of those indices in the E-MOP.

4. IF events are found, THEN check that they have all features of the target event. If any do, return them, and finish.

5. IF E-MOPs are found, traverse them in parallel using this algorithm.

```
                +----------------+
                | select indices | <----------------+
                +----------------+                  |
                   /        \                       |
              none/       +---\--------------+      |
                 /        | traverse indices |      |
         "not found"      +------------------+      |
                       for each index in parallel:  |
                                 /          \       |
               found event  /                \ found E-MOP|
                 +--------/-------+         +-----------+
                 | check features |
                 +----------------+
                      /      \
             no match/        \ match
                    /          \
            "not found"      return event
```

Figure 2-2

2.3 The need for elaboration

In each of the examples above, traversal continued until a unique event was found. Each of those target concepts specified some event feature that was both indexed in the E-MOP and unique to one event in the E-MOP. Thus, we can conclude that retrieval of a target concept which specifies an event feature or combination of features which is both *indexed* and *unique* can be done easily through traversal.

We cannot expect that every question asked, however, will specify indexed unique features. When a target concept specifies an unindexed feature or does not specify a unique combination of features, the traversal algorithm presented above will fail.

Suppose, for example, that we wanted to answer the following question:

(Q2-3) How many times have you met with Begin about the Camp David Accords?

In order to answer this question, all meetings with Begin about the Camp David Accords must be found. Because enumeration of events is not possible, answering this question requires individual retrieval of each appropriate meeting. In order to traverse E-MOPs to find those meetings, the indices to be traversed must be specified.

Meetings with Begin about the Camp David Accords can be reached, in the E-MOP below (repeated from figure (1-5)), by first traversing the branch from MOP1 -- "diplomatic meetings" -- indexing Begin as a participant, arriving at MOP2 -- "diplomatic meetings with Begin". That

E-MOP's topic index with value the Camp David accords can then be traversed, arriving at index point (7) in the tree, where all diplomatic meetings with Begin about the Camp David Accords are organized in MOP4. Alternatively, these meetings can be reached by first traversing a topic index in MOP1 with value equal to the Camp David Accords, arriving at MOP3 -- "diplomatic trips about the Camp David Accords". From that E-MOP, the index specifying Begin as a participant can be traversed, arriving at index point (8) in the diagram above, also an index to MOP4 -- "diplomatic meetings about the Camp David Accords with Begin". Breadth-first traversal, in this case, will arrive at MOP4 in two different ways.

```
                        "diplomatic meetings"

                (1)  (MOP1)
content frame:  the actor is Cyrus Vance
                participants are foreign diplomats
                topics are international contracts
                participants talked to each other
                goal was to resolve disputed contract
differences:      /            \
         participants           topic
        ../ |    \          .../ |  \..
      ./  (2)|    \         /    |    \ (3)
     |   Dayan  Gromyko  SALT    |    Jerusalem
     |       |     |       |     |        |
     |      EV4   EV2     EV2    |       EV3
     |                       (5) |
(4)  Begin  (MOP2)            Camp David Accords (MOP3)
content frame:                  content frame:
  participants include Begin      topic is the CDA
  topic concerns Israel           participants are Israeli
     and Arabs                    specialization of MOP1
  specialization of MOP1        diffs:    |
diffs:      |                             | participants
            | topic               (8)    / \ (9)
 (6)       / \  (7)                 Begin   Dayan
 Jerusalem   Camp David Accords     (MOP4)    |
     |           (MOP4)                      EV4
    EV3
```

Figure 2-3

At this point, the traversal process, as specified above, must abort. There are no additional features specified in the target concept which correspond to indices in the "meetings about the Camp David Accords with Begin" MOP. Thus, no further traversal can be done. Although all meetings between Vance and Begin about the Camp David Accords can be found in MOP4, they are indexed in that E-MOP and can only be retrieved through specification of the differences that index

them.

The problem here is that an E-MOP has been retrieved instead of an event. Individual events from that E-MOP must be extracted from it, but there are no specifications in the target event that can be used to traverse the E-MOP. This happens when a retrieval specification is too general. If a retrieval specification does not specify a unique event, but specifies an entire class of events, then additional event features must be specified in order to traverse deeper into the classification.

Suppose, for example, that MOP4 -- "diplomatic meetings about the Camp David Accords with Begin" -- had the following structure:

```
   "meetings with Begin about the Camp David Accords" -- MOP4

   content frame:  participants include Begin
                   topic is the Camp David Accords
   diffs:         /               \
                 /            included in
                /               /      \
           place            Summit      Diplomatic
          /  |  \          Conference    Trip
    Israel Belgium  USA     (MOP10)      (MOP9)
      |      |    (MOP8)      |             \
     EV7    EV6             place           place
                            /   \          /    \
                      Belgium   USA    Israel  Belgium
                         |       |        |       |
                        EV6     EV8      EV7     EV6
```

Figure 2-4

If possible places for these meetings or the types of events they were part of could be inferred, then actual meetings could be retrieved from the MOP. We call the process of specifying additional features of a target event elaboration.

The processes CYRUS uses for elaborating on a retrieval specification are called instantiation strategies. They use information specified in a target concept and information associated with the E-MOPs the target concept fits into to fill in or better specify target concept features. There are two classes of features instantiation strategies elaborate on: event components and event context. Event components include such features as participants, location, and topic. An event's context includes other episodes which were related to it.

Consider, for example, the following instantiation rule:

"Infer-Country"

To infer the country an event might have taken place in, use the participants' country of residence, country they habitually travel to, or their nationality.

Figure 2-5

Applying this rule to "meetings with Begin about the Camp David Accords" (the target event of (Q2-3)), "Israel" can be inferred as a plausible place for one of these meetings. The index for "location is Israel" can then be traversed in the E-MOP, and EV7 will be found. The set of strategies which infer event components and slot fillers are called _component_ _instantiation_ strategies. Additional component instantiation strategies will be presented in the next chapter.

Besides "location" indices, the "meetings with Begin about the Camp David Accords" MOP specified in figure (2-4) has indices for types of E-MOPs its events were included in. Thus, specification of the kinds of episodes a "meeting with Begin about the Camp David Accords" was part of can also help in retrieving individual events from that E-MOP. The set of instantiation strategies which construct event contexts related to a target event are called _context-to-context_ _instantiation_ _strategies_. These strategies use E-MOP information from E-MOPs the target event fits into to elaborate on contexts related to the target event.

One of these strategies, called "Instantiate Larger Episodes", uses information about the kinds of larger E-MOPs its episodes are normally part of to predict episodes the target event could have been embedded in. Because "diplomatic meetings" often occur during negotiations and summit conferences, and because "diplomatic meetings" are "diplomatic activities" (another E-MOP), which often happen during "diplomatic trips", that strategy can be used to elaborate the target event "meetings with Begin about the Camp David Accords" in the following three ways:

"meetings with Begin about the Camp David Accords which took place during a diplomatic trip to Israel"

"meetings with Begin about the Camp David Accords during a summit conference which included Begin as a participant and whose topic was related to the Camp David Accords"

"meetings with Begin about the Camp David Accords which were part of negotiations with Israel about the Camp David Accords"

Retrieval of each of these can then be attempted. The first will retrieve EV7 by following the indices "included in a diplomatic trip" and then "place is Israel". Retrieval of the second will include traversal of the index "included in a summit conference", and further elaboration will be needed to retrieve an individual event. The third will not be successful in the E-MOP illustrated.

2.3.1 When is elaboration appropriate?

One case when it is appropriate to apply elaboration is when a target event does not describe a unique event, but refers to a set of events, as in the examples above. In that case, additional features that could describe individual events are inferred.

Traversal must also fail if a target concept specifies features which are not indexed. Suppose, for example, that we wanted to answer the following question:

(Q2-4) Where did Vance go on diplomatic trips last April?

The target event of this question is "diplomatic trips by Vance last April". Suppose that the following "diplomatic trip" MOP were the one being traversed to answer this question:

```
                          "diplomatic trip"

content frame:  the actor is Cyrus Vance
                for the purpose of negotiations
                to a foreign country
                the first and last events are flights
                after arrival, there is a welcome ceremony
                includes diplomatic activities
                lasts a few days
                sleep in "visiting dignitary" accommodations
            /   includes a meeting with the head of state
           /              |                  \
       place              |                   \
     /   |  \          purpose             duration
    /    |   \         / |   \              /     \
  the    | Africa     /  |    \....     1 day   two weeks
Mid East |   |       /   |         \      |        |
(MOP1)   |  EV2     /    |          \    EV2      EV1
         |     discuss  negotiate  discuss
      Europe   detente  Mid East   economic
      (MOP2)   (MOP3)   peace      relations
                        (MOP4)      (MOP5)
```

Figure 2-6

The only feature the target event specifies is a time, but this E-MOP has no indices for time. The same elaboration rules that can be used to elaborate additional features when none are available can be used to _transform_ inappropriate indices into appropriate ones. By recalling what kinds of important international problems were going on last April, for example, possible places and purposes for diplomatic trips could be inferred. Suppose, for example, that one of the international problems going on last April had to do with setting up the Rhodesian government. It would be plausible in that case, to hypothesize that Vance spent some time last April in Rhodesia, and to attempt retrieval of a trip to Rhodesia last April. Thus, the feature "place is Rhodesia" could be inferred and traversal attempted. This, however, would still not be sufficient in the E-MOP above. Its place indices are parts of the world, not countries. Thus, elaboration would be used again to transform "place is Rhodesia" into "place is Africa". That index could be followed, finding EV7, and its description could be checked to see if it had occured last April.

We call the process of transforming an inappropriate index into one that can be used for traversal _index fitting_. There are three ways a feature can be "fit" to an E-MOP.

1. A given feature can be specialized

2. A given feature can be generalized

3. Given features can be used to derive other possible features

Transforming the feature "place is Rhodesia" to "place is Africa" as in the example above is an example of index generalization (2). Using the time specification to infer a purpose or place in the example above is an example of using given features to derive other possible features (3). This type of index fitting is equivalent to the elaboration done when a target specification is too general.

Feature specialization (1) can be done when a specified feature is too general to be traversed in the E-MOP, but when some of its more specific instances might be indexed. Suppose, for example, that we wanted to answer the following question using the "diplomatic meetings" MOP in Figure (2-3):

(Q2-5) How many times have you talked to Begin about Arab Israeli concerns?

This question has target concept "meetings with Begin about Arab Israeli concerns". Using the "diplomatic meetings" MOP in figure (2-3), traversal can be directed to MOP2 "meetings with Begin". The problem at this point is that the remaining feature that has not yet been traversed -- "topic concerns Arabs and Israel" -- is too general to use in traversing the E-MOP. In fact, it is one of the norms of that E-MOP. More specialized topics involving the Arabs and Israel are used as indices in the E-MOP, however. It would thus be appropriate to specialize "topic concerns Arabs and Israel", and to infer particular

Arab-Israeli concerns that might be indexed.

In summary, there are two appropriate circumstances when elaboration should be used:

1. when a target concept is too general for retrieval of a unique event, i.e., for *plausible index generation*

2. when a target concept specifies features that are not indexed, i.e., for *index fitting*

2.3.2 Guiding elaboration

Elaboration of a context involves specifying or elaborating some aspect of the context, often filling in a place, participant, or time specification. Instantiation strategies correspond to reasoning rules for filling in contextual details. After using these strategies to fill in details during retrieval, traversal can be attempted again on the better specified context. In this way we can imagine a dialog going on between elaboration processes and traversal processes. Recall that in the second step of the traversal process, if there are no indices specified for traversal, traversal fails. If, instead at that point, traversal returns specifying that it needs more information, then elaboration can take over, and traversal can be continued using the new specifications. We can envision that process as follows:

```
                    traversal/elaboration

                 +----------------+
                 | select indices | <------------------+
                 +----------------+                    |
                   /          \  done                  |
         "more info      +---\---------------+         |
          needed"     -->| traverse indices |          |
    +-------/-----+   /  +------------------+          |
    | elaboration |  /   found /          \            |
    +-------------+ /    event/            \found E-MOP|
       /     \     /   +------/----------+  +----------+
   failed    +---+     | check features |
     /      success    +----------------+
"not found"                   /    \
                       no match/    \ match
                             /        \
                       not found"   return event
```

Figure 2-7

How can elaboration be guided? Only those features which are indexed should be elaborated. It would only be appropriate to elaborate on a target event's location, for example, if the E-MOP being traversed indexed "locations" (or if the location was needed to elaborate on some other aspect).

If E-MOPs have available the types of features they index, then when the traversal process aborts, it can specify the E-MOP it stopped at and the types of indices in that MOP. Only instantiation strategies associated with those types of indices need be applied. Note that this does not mean that values for every feature indexed in an E-MOP must be available, but only that the nature of the indices be known. Instantiation rules associated with each appropriate type of feature will generate values for the indices.

This implies that E-MOPs have a two-tiered indexing structure. The first tier has the types of features which are indexed. That tier must be enumerable. The second tier is indexed off of the first and has values for each type of feature. It does not need to be enumerable since instantiation strategies will provide values. Thus, traversal of the second tier can be done through direct access.

Using the E-MOP in figure (2-6), to answer (Q2-4), for example, traversal would return with the information that the MOP had "place", "purpose", and "duration" indices. Elaboration rules associated with each of those would be applied to compute actual indices. In this way, memory organization guides choice of features to be elaborated, limiting choice of instantiation rules.

We can amend the traversal strategy presented above as follows to include elaboration.

Traversal/Elaboration

1. Select possible indices for the target event based on its specified features and their differences from the norm of the E-MOP being traversed.

2. IF there are no appropriate indices, THEN return (a) the E-MOP being traversed and (b) the nature of its indices. Call instantiation strategies to elaborate on the target concept and attempt traversal/elaboration of the newly-elaborated target.

3. ELSE follow all of those indices in the E-MOP.

4. IF events are found, THEN check that they have all features of the target event. If any do, return them, and finish.

5. IF E-MOPs are found, traverse them in parallel using this algorithm.

Figure 2-8

This algorithm describes the diagram in figure (2-7). The significant change in this algorithm over the initial traversal algorithm is that it specifies that instantiation strategies be applied to elaborate an underspecified target concept. Step (2) above is equivalent to a message from memory saying "I don't have enough information to traverse deeper in my E-MOP structures. If you give me the following types of information, I may be able to search deeper". When traversal returns that message, instantiation rules can fill in necessary details, and traversal can be attempted again. This dialog continues until appropriate events are found or until no more elaboration can be done. In this way, memory organization guides retrieval.

2.3.3 Types of elaboration

There are three types of elaboration:

(1) feature specification

(2) feature inference

(3) feature enumeration

In feature specification, a specified feature of the target event is better specified. In feature inference, a feature not specified in the target event is inferred. In feature enumeration, a set of possible values for some feature is computed.

Consider again the following question:

(Q2-6) Where did Vance go to on diplomatic trips last April?

We explained above that in order to answer this question, features indexed in the "diplomatic trips" MOP had to be inferred. We suggested using information about what was going on in the world last April to infer possible trouble spots where Vance might have been, and made the inference that Vance might have been in Rhodesia. In this way, the following transformation of the target event was made:

```
target event:
          "diplomatic trips last April"

                     | |  infer-location
                     \ /
modified target:
       "diplomatic trips to Rhodesia last April"
```

A new feature (location) which had not been previously specified in the target event was inferred. One task elaboration can do, then, is to infer event components not previously specified.

In attempting retrieval of the new modified target event "diplomatic trips to Rhodesia last April", retrieval failed because the place index in the "diplomatic trips" MOP specified parts of the world and not countries. To traverse the E-MOP, elaboration had to be applied to generalize "Africa" from "Rhodesia". The following target event transformation was done:

```
target event:
          "diplomatic trips to Rhodesia last April"

                     | |  infer-part-of-world
                     \ /
modified target:
       "diplomatic trips to Africa last April"
```

In this case, a feature which was already specified in the target event (location) was better specified to correspond to E-MOP indices through generalization of its value. Both generalization and specialization of particular features can be used to better specify an already-specified event feature for traversal.

Often, specification of features involves enumeration of possibilities. Both partially-specified or previously unspecified features can be enumerated. Consider, for example, the following question:

(Q2-7) How many of the states in the United States has Vance been to?

If a map of the United States could be reconstructed and each state named as that reconstruction was going on, then the 50 states could be named and retrieval of events in each state attempted. Enumeration of this type has been referred to as "map search" by Schank (1980). Map or chart application, or well-ordered enumeration, can be helpful in specifying possible features of a target event. Well-ordered enumeration is a useful type of elaboration.

2.3.4 Elaboration in human protocols

Let us return to some of the questions we asked people and explain their answers. The following two protocols illustrate traversal and application of component-instantiation strategies for elaboration in answering "Name all the museums you've been to".

> What kinds of museums have I been to? Art museums, there have been a lot of them. I'll come back to them. Science museums -- there's a good one in Philadelphia, the Franklin Institute, there's one in Boston that I've been to, and there's an air and space museum in Washington that I've been to. Wax museums -- there's one in London -- Mme. Taussaud's. ...

> I know I've been to a lot of museums in Europe. I've been to England, and I went to a number of museums there -- some in London -- the British Museum, the National Gallery, and a few smaller galleries, ... I was at a museum in Brighton -- the Royal Pavilion. I've been to museums in Paris -- the Louvre and some smaller ones. In Rome, I've been to .... In Naples, to .... In Florence, to ....

In the first protocol, the respondent seems to be first elaborating types of museums, then elaborating places where those museums might be. Thus, we can imagine that component instantiation was initially used to supply a museum type. Elaboration continued for each type of museum by enumerating places.

We can explain this protocol using the traversal/elaboration process described above. In recalling museum experiences, the traversal process returned a message saying "I need more information. The type of museum would be useful". At that point, component-instantiation rules were applied to enumerate museum types and traversal continued. As traversal was attempted for each type of museum, the traversal process could have returned the message "I need more information, place would be useful". Component-instantiation strategies could then be used to enumerate places and retrieval of each newly specified target could be attempted.

The second protocol was from a person familiar with Europe. He remembered museums he had been to in Europe by enumerating places in Europe. We can imagine him mentally walking through a map of Europe stopping at each familiar or important point and attempting to use it to further specify the target event.

2.4 Implications of traversal/elaboration

In the traversal algorithm described, indices must be specified in order for an E-MOP to be traversed. This requirement has a number of important consequences. An event cannot be retrieved unless a suitable discriminating index for it can be inferred. Because the process does not allow enumeration and search of all E-MOP indices, but requires determination of plausible indices, all events matching a particular specification will not always be found. Advantages of this limitation, however, far outweigh that disadvantage.

The major advantage of this algorithm is that it directs search only to relevant places in an E-MOP. Thus, if a MOP has hundreds of indices organizing thousands of events, they do not all have to be searched. Only the ones directly related to the targeted context specification are traversed.

The second advantage of this algorithm is that it allows generalized information to be retrieved in exactly the same way events are retrieved. Thus, retrieval for a question such as "When you go to Europe, where do you usually stay?" will proceed exactly as for "Last time you were in Europe, where did you stay?" To answer the second question, the particular trip to Europe would be retrieved by the traversal process (as described above), and its context would then be searched for where the actor stayed. In traversing memory to answer the first question, an E-MOP corresponding to "trips to Europe" would be returned by the traversal process in exactly the same way a particular event is retrieved. Generalized information associated with that MOP would then be used to answer the question.

2.4.1 Recognizing whether or not an event is in memory

When a target event cannot be found, how can we know whether or not it is actually in memory? An event that cannot be found either is not in memory or is in memory but not retrievable because the proper indices cannot be specified. There are two ways of evaluating whether or not an event is in memory. In the first step of the traversal algorithm, possible indices for the target event are selected. The same process that chooses features of an event to be indexed when it is added to memory is used during retrieval to choose features that would have been indexed if the target event were in memory. A procedure for deciding that an event is not in memory will take a target's possible indices into account.

If some possible index of the target event is unique in an E-MOP, and the event that it indexes does not match the target event, then we can assume that the target event is probably not in memory. Suppose, using the following E-MOP (repeated from figure (2-3)), that we wanted to answer the follcwing questions:

(Q2-8) Have you met with Gromyko about the Afghanistan invasion?

(Q2-9) Have you met with Dayan about the status of Jerusalem?

```
                       "diplomatic meetings"

              (1)  (MOP1)
content frame:  the actor is Cyrus Vance
                participants are foreign diplomats
                topics are international contracts
                participants talked to each other
                goal was to resolve disputed contract
differences:      /             \
         participants          topic
       ../  |    \         .../  |  \..
      ./  (2)|    \          /    |     \  (3)
      |    Dayan  Gromyko  SALT   |     Jerusalem
      |      |      |       |     |        |
      |     EV4    EV2     EV2    |       EV3
      |                       (5) |
(4)  Begin  (MOP2)           Camp David Accords (MOP3)
content frame:                  content frame:
  participants include Begin      topic is the CDA
  topic concerns Israel           participants are Israeli
     and Arabs                    specialization of MOP1
  specialization of MOP1        diffs:    |
diffs:      |                             | participants
            | topic               (8)    / \ (9)
 (6)      / \  (7)                  Begin   Dayan
 Jerusalem   Camp David Accords    (MOP4)     |
    |            (MOP4)                      EV4
   EV3
```

Figure 2-9

One of the indices that should be traversed to answer this question is the index "has Gromyko as a participant". This E-MOP indexes one meeting with Gromyko (EV2), and that meeting is about SALT. Thus, the index "has Gromyko as a participant" indexes a unique event. Since that event does not correspond to the target event "a meeting with Gromyko about the Afghanistan invasion", we can safely assume that there is no meeting corresponding to that target event. If there were, it would be indexed at that point.

A negative answer to the second question can be derived similarly. Its target concept is "diplomatic meetings with Dayan about Jerusalem". The indices "participants include Dayan" and "topic concerns Jerusalem" can be followed, retrieving EV3 and EV4. If neither match the target specification, we can assume there is nothing in memory corresponding to it.

There is a second heuristic for deciding whether or not an event is in memory. Just because an E-MOP does not index a particular given feature of an event does not mean that the event is not in memory.

However, if an E-MOP has indices of the same type as an index selected from the target, but it has no particular index corresponding to the value of that feature in the target, then we can conclude that the event is not in memory. This heuristic corresponds to what Collins, et. al. (1975) and Collins (1978) call a "lack of knowledge" inference.

Consider, for example, how the following question could be answered using the E-MOP in Figure (2-9).

(Q2-10) Has Weizmann been to any of your diplomatic meetings about the Camp David Accords?

The target event in the first question is "diplomatic meetings with Weizmann about the Camp David Accords". One of its features, appropriate for indexing in that E-MOP, is "participants include Weizmann". Since there are indices for participants in that E-MOP and they index individual people, the MOP must not include an event corresponding to the target event.

Similarly, suppose that an E-MOP "diplomatic meetings about Arab Israeli peace" had the following structure:

```
          "diplomatic meetings about Arab-Israeli peace"

content frame:  topic is Arab-Israeli peace
                underlying topic is peace
                involves Israel and the Arabs
                participants are heads of state
                            |
differences:                |
                      participants'
                      nationality
                        /    \
                       /      \
                   Israel     Egypt
```

Figure 2-10

Suppose we wanted to answer the following question:

(Q2-11) Was King Hussein ever at a diplomatic meeting you attended about Arab-Israeli peace?

According to the E-MOP above, only Israelis and Egyptians have attended "diplomatic meetings about Arab-Israeli peace". Since (a) Hussein is a Jordanian, (b) there are indices for participants' nationalities in the E-MOP, and (c) Jordan is not one of them, we can assume there has never been a meeting with Hussein about Arab Israeli peace. If there had been, there would be an index for it.

It is not always possible to determine with certainty whether or not an event that cannot be found is actually in memory. If all known possibilities for features have been enumerated and retrieval has been attempted on all of them, then there is a good chance that the event is not in memory. If, however, appropriate indices in an E-MOP cannot be completely enumerated, there is no way of knowing if a target event is in memory or not. The assumption can be made that it is probably in memory but can´t be found. Instantiation strategies can be used to infer information that cannot be found directly.

This uncertainty seems to correspond to what Collins (1978) and Norman (1972) have called a "feeling of knowing". When people experience a feeling of knowing, they think, but cannot verify, that a fact is true. They have a "feeling of knowing" that it did happen, but cannot find the appropriate information to verify its occurence. That feeling of knowing can happen when (1) the traversal/elaboration process arrives at an E-MOP which may contain the target event, and (2) elaboration processes are inadequate for specifying indices for the target event in that MOP. When that happens, an answer of "probably" or "possibly" can be given rather than answering "no" or "I don´t know".

2.5 Context construction

In each of the examples above, an E-MOP for traversal was specified in the question. The question "Have you attended a diplomatic meeting about SALT?", for example, specifies the E-MOP "diplomatic meetings" for traversal. The following questions, however, do not directly specify E-MOPs for traversal.

(Q2-12) Who have you talked to about SALT?

(Q2-13) Have you ever discussed the Camp David Accords with Dayan?

(Q2-14) Who is the most famous person you have ever met?

The first two questions are appropriate to ask the CYRUS data base about Vance. The last is a question we have asked a number of people. Suppose we wanted the CYRUS data base to answer the first two questions. CYRUS knows about many different situations in which Vance might talk to other people (in fact, almost any situation could apply), but it does not have an E-MOP for "talking to people". How, then, can it search memory to answer these questions? Although Vance could talk to somebody during any of his activities, it would not be appropriate for CYRUS to search every one of its E-MOPs to find discussions about SALT or the Camp David Accords.

In order to traverse memory, it is first necessary to specify which parts of memory ought to be searched, i.e., which E-MOPs are appropriate for traversal. The task, then, is to choose appropriate E-MOPs. CYRUS chooses appropriate E-MOPs for traversal by looking at the kinds of situations associated with each of the given question components. In answering (Q2-12), it can use information about SALT to choose an E-MOP.

Because SALT is an international contract, CYRUS assumes that Vance would have discussed it at a political meeting of some sort -- either with a foreign dignitary or with a United States official. Using that information, CYRUS knows to search its "diplomatic meetings" and "consultations" E-MOPs to find an answer. In answering (Q2-13), CYRUS uses knowledge about both "the Camp David Accords" and "Dayan" to decide that the correct E-MOP to search is "diplomatic meetings".

In general, in order to search memory, a _context for search_ must first be set up. A context for search specifies a memory category to be searched. CYRUS' organized memory categories are event categories, or E-MOPs. Thus, a context for search in CYRUS must include a specification of an E-MOP for traversal. The strategies in CYRUS which construct contexts for search when none is specified in a target concept are called _component-to-context instantiation strategies_. These context instantiation strategies make use of event information associated with each given question component to infer plausible E-MOP specifications.

In answering (Q2-13), two context-instantiation strategies are used. The strategy "Infer-from-Topic" is applied to the "Camp David Accords" to infer a "political meeting" context, and the strategy "Infer-from-Participants" is applied to "Dayan" to infer a "diplomatic meeting" context, a refinement of "diplomatic meeting". "Infer-from-Topic" makes use of topic information to infer an E-MOP, while "Infer-from-Participants" makes use of participant information. Those strategies, along with other context-instantiation strategies, will be explained in detail in the next chapter.

What kinds of information must be available to construct contexts for search? Since context-instantiation strategies infer event contexts from event components, we must assume that each of those components has event information associated with it. To answer the questions in these examples, we must assume that person classifications and topic classifications have event information associated with them. In particular, the topic classification "international contract" must have associated with it the context "political meeting", and the person classification "foreign diplomat" must have associated with it the event context "diplomatic meeting".

The retrieval process, including context construction and traversal/elaboration, must include at least the following:

The Retrieval Process

1. IF the target event does not specify an E-MOP for traversal, THEN apply component-to-context instantiation strategies to infer a context for search.

2. Traverse memory searching for that search context (using the traversal/elaboration process specified above).

3. IF an appropriate answer has not been found, go back to step 1, attempting to construct a new context.

Figure 2-11

Context-instantiation strategies construct contexts for search when an episodic context is not specified. To do that, they use event information associated with components of the target event. Traversal/elaboration can then proceed as described above.

2.5.1 Human protocols and context construction

When we asked people to recall the most famous person they had ever met (Q2-14), we found that people used a retrieval process similar to that just described. The following two protocols typify the answers we got to "Who is the most famous person you have ever met?"

> Since I've been at Yale, I've met famous people at the Chubb fellow seminar series. I met President Carter at one of those seminars, before he was president, ...

> First I thought how somebody could be famous, and politics was the first thing I thought of. Then I thought about circumstances where I could have met a famous politician. I searched political experiences I have had -- mostly political rallies I participated in and experiences campaigning for candidates. I remembered that I had met McGovern. But since you said 'most famous', I went on to think of other famous people I might have met. Next I thought of entertainers, and how I could have met them ... Then I thought of famous scientists, and places I could have met scientists. ...

What people seemed to be doing was to construct contexts during which they would have met famous people. The person who gave the first protocol could have used the strategy "Infer-from-Participants" to infer a context he had associated with "famous people". We can explain the second protocol as the result of a failed initial attempt to apply that

strategy. We can assume that person had no event contexts associated with "famous person", or he would have attempted retrieval of those contexts. Because the only specification given in the question was participants, that was the only choice for elaboration. By specializing "famous person", concepts with associated event contexts (e.g., politician, entertainer) could be inferred, and "Infer-from-Participants" could be applied to each of those. Our explanation of why that person enumerated types of famous people, then, is so that he could then search individually for contexts associated with each.

It seemed that people applied context-instantiation to construct and then search for political rallies, participating in political campaigns, and going back-stage after the theater, to name just a few. Each of these is an instance of a context in which a particular kind of famous person could have been met. Each is an instance of inferring an E-MOP for traversal from participant information available in a target concept. In order for this to happen, "politician", as a memory concept, must have associated with it the fact that they normally take part in political rallies and campaigns, and "entertainer" must have associated with it the fact that they can be found in plays.

Using these strategies, we can predict a plausible reasoning process for answering the following question about Cyrus Vance:

(Q2-15) What heads of state have you met?

Vance meets people in many different types of situations every day -- at meetings, parties, conferences, on the street, in his office, etc. Furthermore, meeting heads of state is probably not very different from meeting any other person (for Vance). Thus, it doesn't seem like "meeting people" or "meeting heads of state" would be good E-MOPs for Vance. This question can be answered, however, by applying "Infer-from-Participants" to infer E-MOP-related situations in which he would have met heads of state.

In order for that to happen, we assume that "heads of state" or "diplomats" has associated with it knowledge about the kinds of activities they are most often involved in with him. Heads of state are involved in "summit conferences". They are also diplomats, and diplomats are normally involved in "diplomatic meetings" and "negotiations". Thus, by applying "Infer-from-Participants" first to "head of state" and then to "diplomat", "summit conference", "diplomatic meeting", and "negotiations" situations involving heads of state can be inferred and their E-MOPs traversed. Thus, the following reasoning sequence is one way of answering this question.

1. Heads of state are often involved in summit conferences. Have I been to any recently? ...

2. Heads of state are diplomats. What kinds of episodes am I involved in with foreign diplomats? Negotiations and diplomatic meetings. What countries have had problems that I've negotiated? Who did I meet with during those negotiations? Have I been to any meetings recently? Who was at those meetings?

2.6 Alternate-context search

The traversal/elaboration process must halt when no further elaboration can be done. Often, when appropriate elaboration cannot be done, however, there are still additional events to be found. Consider the following example:

(Q2-16) Enumerate all of Vance's meetings with Gromyko.

Suppose that memory held a large number of "diplomatic meetings" with Gromyko, some in the United States, some in Russia, and all to discuss various aspects of SALT. Because they are so similar, it is not likely that elaboration by itself would be enough to specify unique event descriptions for retrieval of all of them. If traversal/elaboration were the only retrieval process, then many of these meetings would not be retrievable.

Because events happen in the context of other events, and because they refer to those related events, an event can be found by finding an episode it was related to. When a related event is found, its context can be searched for the target. Thus, to search for a meeting with Brezhnev, it might be appropriate to recall a summit conference which would have included Brezhnev and to retrieve a meeting with Brezhnev from the summit conference's sequence of events. Since summit conferences are less common than diplomatic meetings, they might be easier to retrieve.

Search for alternate episodes can aid retrieval in answering (Q2-16). Those meetings were all related to different episodes. Some were parts of diplomatic trips, some summit conferences, some called for crisis reasons, some enabled a breakthrough in negotiations, etc. If diplomatic trips, summit conferences, crisis situations, and negotiations breakthroughs related to those meetings could be found, the meetings themselves could be retrieved.

Context-to-context instantiation strategies are used to construct alternate contexts for search. These are the same strategies used during elaboration to elaborate on episodes related to a target event. In order for possible related contexts to be inferred from a target event, E-MOPs must specify both the types of episodes (i.e., E-MOPs) they are often related to, and how those episodes are related, i.e., how their roles correspond to those of the E-MOP. "Diplomatic meetings",

for example, specifies that meetings are instrumental to "negotiations". It also specifies the types of negotiations they are instrumental to. That specification includes the information that the topic of related negotiations episodes is usually a contract involving the sides of the meeting and related to the meeting topic, that the sides of the negotiations correspond to both the countries of nationality of the meeting participants and the sides of the contract being negotiated, and that the arbitrator might be a person from a country neutral to the conflict being negotiated and could be one of the meeting participants.

When a particular meeting is being searched for, the negotiations episode it would have been part of is inferred by applying a context-instantiation strategy to the specifications of the target meeting and the generalized information from "diplomatic meetings". To recall a meeting between Vance and Begin by searching for negotiations it could have been part of, context-instantiation rules will use "diplomatic meeting" information to infer a negotiations episode involving Israel as a side in the negotiations, probably with topic Arab-Israeli peace (since that is the current most crucial topic of negotiations involving Israel), and with Vance as the arbitrator representing the United States.

2.6.1 Guiding alternate-context search

Search strategies direct search of memory by directing application of Instantiation strategies and traversal of memory to find alternate contexts. Once an alternate context is retrieved from memory, its context must be searched to find the target. Thus, if memory were searched for "crisis situations involving Russia" to find a "diplomatic meeting with Gromyko", then events resulting from the crisis situation would be checked to find the possible related meeting with Gromyko. In searching memory for trips in order to find visits to museums, the sequence of events of the trips found must be searched for actual museum experiences.

Search strategies, then, must direct search within a particular episode for the target event. During retrieval of diplomatic meetings, search strategies guide construction of related contexts for retrieval (e.g., negotiations, summit conferences, crisis situations, diplomatic trips), trigger appropriate context instantiation strategies to construct those contexts, traverse memory to retrieve them, and if one is found, direct search within that context for the original target event -- diplomatic meetings. Search strategies, then, have 4 steps:

1. choose a context to be constructed or elaborated

2. call the appropriate instantiation strategy to construct or elaborate that context

3. retrieve that context from memory

4. search for the target event in the surrounding contexts of the events retrieved

In searching for "diplomatic meetings with Gromyko", a search strategy which searches for episodes an event could have been part of would (1) call the context instantiation strategy "Infer-larger-episodes" to construct contexts for summit conferences about a Russian-American concern, diplomatic trips to Russia, and negotiations concerning a Russian-American contract, (2) traverse memory searching for each of those, and (3) search the sequence of events of each episode found for an appropriate diplomatic meetings.

Search strategies are used to guide search for event contexts related to a target event. In doing that, search strategies direct search for "what must have happened" or "what may have happened" if the target event had taken place. If an event that must have happened along with a targeted event can be found, its specification in memory might refer to the target.

Why should searching for alternate contexts aid retrieval? An alternate context might be less common than a target event, and thus need less of a specification in order to be retrieved. Summit conferences happen less often than diplomatic meetings do. Although a specification "diplomatic meetings about SALT" might describe many meetings, the specification "summit conferences about SALT", which can be derived from that, describes only a small number of events.

Alternate-context search can also aid retrieval of a target event if the alternate context can be better specified with respect to the E-MOPs it is indexed in than the target event can. "Diplomatic meetings with Begin" might be an ambiguous target event. Although "diplomatic meetings" might often happen as a result of "crisis situations", if there had only been one crisis situation which involved Israel, then it would be better specified in its E-MOPs than "meetings with Begin" are in "diplomatic meetings". Thus, it would be more retrievable and aid retrieval of at least one meeting with Begin.

Where does alternate-context search fit into the retrieval process? The first step in retrieval is to construct a search context, i.e., an event description specifying an E-MOP to be traversed. Next, traversal/elaboration is done. Traversal/elaboration corresponds to looking directly for the target event. Since that should always come before search for related events, alternate context search should come after traversal/elaboration when traversal/elaboration fails. The following process, then, describes reconstructive retrieval.

The Retrieval Process

1. IF the target event does not specify an E-MOP for traversal, THEN apply context instantiation strategies to infer a context for search.

2. Traverse memory searching for that target event (using the traversal/elaboration process specified above).

3. If an appropriate answer has not been found, THEN apply search strategies to search for alternate contexts related to the target.

4. IF an appropriate answer has not been found, THEN go back to step 1, attempting to construct a new context.

```
       +---------------------+
       | construct a context | <-----------+
       +---------------------+             |
                 |                         |
                \ /                        |
     +----------------------+              |
     | traversal/elaboration |             |
     +----------------------+              |
         /          \                      |
     succeed        fail                   |
        /          +--\---------------+    |
   return          | alternate context |   |
   events          |  search           |   |
                   +-------------------+   |
                       /       \           |
                   succeed      \ fail     |
                     /           \---------+
              return events
```

Figure 2-12

2.6.2 Human protocols and alternate context search

Searching for alternate contexts is something that people seem to do quite often in searching memory extensively. In answering the question "Name all the museums you have been to", we found that people often attempted to recall trips that had been on or visits from out-of-town friends to remember all of those experiences. The following protocol typifies that type of search.

> "Let me see. ... What other museums have I been to? The last time I was in England, I went to a bunch of museums in London -- the British Museum, some gallery whose name I can't remember, and Mme. Tussaud's Wax Museum. And I also went to some palaces there that were museums -- the Royal Pavilion in Brighton and some other palace in London, um,... it was called Hampton Court. When I went on my first trip to France, ... I once went on a trip around New York state, and I went to the photography museum in Rochester and the Corning Glass Museum. During my trip to California, ..

This person seemed to be applying search strategies to construct and search for particular trips to Europe, searching the sequence of events of each trip for museum experiences, then doing the same for excursions around the United States.

Taking these strategies into account, we can predict how somebody like Cyrus Vance might answer the following question.

(Q2-17) Have you been sightseeing in Dubai?

We could imagine any of the following answers:

> "Why would I have been in Dubai? It must have been to talk about Mid East peace or to negotiate the oil situation."

> "Have I ever been to Dubai? Well, I've been to the Mid East a number of times, mostly to negotiate. Maybe I was in Dubai during one of those trips."

> "I know I've been to Dubai. Does it have any distinctive places where they might have taken me sightseeing? Perhaps the oil fields, or archaeological ruins, or a mosque ..."

In the first, we imagine him searching for reasons why he might have been in Dubai and gone sightseeing. In the second, we imagine search for trips the sightseeing could have been part of. In the third, we imagine enumeration of types of sightseeing experiences.

2.7 Summary -- the entire process

Retrieval includes (1) choosing a context for search, (2) constructing and attempting to traverse memory for that context, (3) elaborating on the context, (4) traversing memory again for each elaboration, (5) inferring and constructing alternate contexts, (6) traversing memory to search for them, and (7) checking their episodic contexts for the initial target concept.

In order for traversal to happen, an E-MOP context must be specified. When no E-MOP specification is given, contexts are constructed from available event components. When poor contexts are given, better contexts are constructed from the given context -- either by better specifying it or by constructing a new related one. When traversal fails, search for related contexts can aid retrieval.

In addition to traversal, then, retrieval must include at least the following:

1. choosing an E-MOP for traversal when there is none specified in a target concept
2. specification of a target concept by filling in hypothetical features
3. specification and search for episodes possibly related to the target concept

Retrieval strategies are the processes that do the retrieval tasks listed above. Retrieval strategies direct memory search and construct new contexts for search. There are two types of retrieval strategies -- processes for construction and further specification of contexts for search, and processes that guide that context construction, or determine what to search for next. Those which direct search, we call _Search strategies_. Those which construct contexts are called _Instantiation strategies_. Instantiation strategies are used to construct contexts for search and to elaborate or flesh out partially specified specifications of contexts.

Applying retrieval strategies is an integral part of memory retrieval and event reconstruction. Strategies serve a number of functions.

1. Strategies can be used to better specify contexts for retrieval by filling in details and providing cues.

In searching for museum experiences, descriptions of museum experiences can be better specified to include where the activity took place and what type of museum it was.

2. Strategies can be used to hypothesize contexts for search when none are explicitly available in a retrieval specification.

In searching for famous people they had met, people searched for the kinds of situations where they could have met famous people (e.g., political rallies, scientific conventions).

3. Strategies can be used to choose contexts for search from already-specified events.

Museum experiences can be found by recalling trips.

4. Strategies can be used to search within specified contexts to find a retrieval key.

After recalling a trip to find a museum experience, the context of each trip must be searched for the museum experiences it included.

Strategies make use of event information associated with E-MOPs, person stereotypes, roles, locational specifications, and other types of memory structures. Strategies can be thought of as general inference rules that relate these entities to each other. Instantiation strategies can be understood as "in the context X, if entity Y is present, there is a strong possibility that entity Z will also be present," thus allowing dense descriptions to be constructed and filled in. Search strategies can be understood as "to find instances of X, instantiate and find instances of the things that relate to X in manner Y, and then search those to find X." Thus, Search strategies direct both application of Instantiation strategies and search for episodic contexts. All retrieval strategies make use of predictions of how different entities in memory co-occur and are causally and temporally related. In the next two chapters, we will show particular strategies, how they work, and the kinds of information they use.

CHAPTER 3

Instantiation Strategies

3.1 Introduction

(Q3-1) Mr. Vance, has your wife ever met Mrs. Begin?

The most interesting feature of the question above is that it does not specify a context for search, i.e., it gives the answerer no idea about where in its memory to look or what kinds of situations to search for. People are faced with questions like this every day, and they manage to figure out the kinds of contexts they should search for. They know enough about the contents of their memories and the kinds of activities they've taken part in to direct memory search. How can a question such as this be answered from a memory which (1) keeps track of Vance's but not his wife's activities, and (2) has no category for "meeting people"? This chapter will explain in detail the strategies that allow retrieval from incomplete specifications to happen.

Instantiation strategies are constructive retrieval strategies. Their purpose is to construct and elaborate on context specifications. As discussed in chapter 2, there are two types of instantiation strategies: *Component-instantiation* strategies, which elaborate on a given target specification by inferring additional components, and *Context-instantiation* strategies, which construct contexts for search. Contexts can be constructed in two ways: Component-to-context instantiation strategies construct contexts from component specifications in a target concept. Context-to-context instantiation strategies infer contexts related to a given context.

Thus, instantiation strategies have three purposes. They are used (1) to fill in unspecified aspects of specified events, (2) to construct event contexts when one is not available, and (3) to construct contexts related to a given context. If the target event is a meeting between Vance and Begin, for example, instantiation strategies can infer that there was a negotiations episode going on, that the topic of the meeting was probably something related to Arab-Israeli peace, and that the

meeting could have taken place in either Israel or Washington. Context-Instantiation strategies would infer the negotiations episode, while Component-Instantiation strategies would infer the place and topic of the meeting. Furthermore, if it were known that the meeting took place in Israel, then Context-Instantiation strategies could infer that the meeting took place during a diplomatic trip by Vance to Israel. Conversely, if the meeting were in Washington, those strategies could be used to infer a diplomatic trip by Begin to the United States.

The types of inferences made by instantiation strategies are used in three different ways. The first two, part of the retrieval process, have been described in chapter 2. First, instantiation strategies are used for elaboration during the traversal process. Second, the inferences made by instantiation strategies are used in applying search strategies.

The third use for these strategies, not previously mentioned, is in reconstructing probable details of events. If asked, "Why did Vance and Begin meet last week?", even if no actual reasons are known, content frame information and instantiation strategies can be used to answer "probably to negotiate Egyptian-Israeli peace". The meeting between Vance and Begin mentioned in the question would first be retrieved using the retrieval process described previously. Instantiation strategies can then be applied to the event found to retrieve its reason.

In a related way, these strategies can be used to predict the kinds of things to look for in memory to reasonably answer a question. In answering "why", for example, as above, possible reasons for the event occuring can be constructed using instantiation strategies, and memory can be searched specifically for those. If those reasons were found, a reasonable answer would have been found, If not, it may still be possible (as above) to use those defaults to answer "probably ....". In other words, instantiation strategies are reasoning rules which use generalized knowledge to aid search for episodes. In a way, these strategies can be thought of as the generation rules for constrained "generate and test" (Newell, 1972). The remainder of this chapter will present the rules.

3.2 Context construction

The rules CYRUS uses to infer a context when none is available are called *component-to-context instantiation* rules. These rules use contextual information associated with available event components to infer contexts for search by retrieving standard contexts associated with the objects and people referred to in the query. Any type of contextual component can be used to retrieve a standard context, including time periods, participants, and locations. Thus, there are context instantiation rules associated with each type of event component. CYRUS has context instantiation rules associated with participants, locations, topics, and time contexts.

3.2.1 Context construction from participant specifications

Contexts can be inferred from particular participants or from participant descriptions. Suppose, for example, that the only situation in which John had contact with Mr. Smith, his butcher, was in Mr. Smith's butcher shop. In that case, a context associated directly with Mr. Smith (buying meat in his store) could be inferred if John were asked the question "When was the last time you talked to Mr. Smith?" If, on the other hand, John did not have an event context associated with Mr. Smith, but did know Mr. Smith was a butcher, he could use information associated with "butchers" to infer a context for that question. The following two context-instantiation rules account for context construction in both of those situations.

Infer-from-Participants

1. If particular participants, such as people, groups, or organizations, are specified, and if any has a particular event context associated, predict those characteristic events associated with each that could also include other already-specified components.

2. If classes of people, groups, or organizations (e.g., "doctors") are specified, predict characteristic events associated with each of them that could also include other already-specified components.

Figure 3-1

Notice that according to both of these rules, the event contexts suggested must be such that they could include other specified event components. This is a requirement of all context instantiation rules. Thus, if Dr. Jones is John's doctor and also belongs to the same swim club John belongs to, then in using this strategy to infer a context for "Did you see Dr. Jones at the pool yesterday?", the context of going to the swim club should be instantiated, and not a context of an "office visit", although that is associated with doctors. After choosing a possible context using these strategies, the context must be double-checked for correlation with other specified components. Because the place specification in the question above contradicts the "visit to the doctor" context for seeing Dr. Jones, it cannot be used.

CYRUS uses this strategy to answer questions such as "When was the last time you talked to a reporter?", "What heads of state have you met?", and "When was the last time you talked to Begin?", i.e., questions which do not specify an event context but which do specify event participants. When asked the question, "What heads of state have you met?", for example, it infers that it should look for "diplomatic meetings". It does that because it knows that "diplomatic meetings" are characteristic events associated with "heads of state" (and other

diplomats). The following is a protocol of CYRUS answering that question:

```
Enter next question:

>What heads of state have you met?

The question is:
((ACTOR HUM1 IS (*PROX* VAL G0749)) TIME G0751)
The question type is "concept completion"
The question concept is:
((ACTOR HUM1 IS (*PROX* VAL G0749)) TIME G0751)
  applying INFER-FROM-PARTICIPANTS to RT-STATE-HEAD
     inferring a diplomatic meeting context
The inferred question concept is:
((<=> ($MEET ACTOR HUM1 OTHERS G0749)) TIME G0751)

searching memory for question concept
  searching directly for $MEET
   ...

Anwar Sadat, President Assad, King Hussein, Hassan, and
Menachim Begin.
```

If CYRUS is asked "What heads of state have you talked to?", it would apply the same rule to infer the same "diplomatic meeting" context inferred above. Talking to someone does not imply an event context. Thus, in this question, like the one above, the specification "heads of state" is used to infer an event context. If, on the other hand, CYRUS is asked "What heads of state have you dined with?", it will not apply these rules, since the question already implies the event context "dining".

3.2.2 Construction from topic specifications

Participants in an event are not always specified. Sometimes, only a topic of conversation is supplied. As with participants, topics of conversation can be used to infer event contexts. In answering "When did you first hear about the Boston Tea Party?", some people infer a context of a high school history class. The Boston Tea Party is a historical event, and the event contexts they have associated with historical events as topics are classroom situations. The following rule can be applied to infer event contexts from specified topics of communication:

Infer-from-Topic

If topics of discussion or classes of topics of discussion are specified, and if any of them have communicatory event contexts associated with them, predict appropriate communicatory events associated with the topic or class of topic.

Figure 3-2

CYRUS uses this rule to answer questions such as "Who have you discussed the Camp David Accords with?" and "Have you ever talked to anyone about Rhodesian independence?" Both the Camp David Accords and Rhodesian independence are international concerns, and event contexts associated with international concerns as event topics are political meetings. Thus, in retrieving people Vance has discussed the Camp David Accords with, CYRUS infers that the discussions would have been at political meetings (as opposed to parties, speeches, etc.), and searches for those types of meetings.

```
Enter next question:

>Who have you discussed the Camp David Accords with?

The question is:
((ACTOR HUM1 <=> (*MTRANS*) MOBJECT
    (*CONCEPTS* CONCERNING CNTRCT1) TO (*?*)) TIME G1380)
The question type is "concept completion"
The question concept is:
((ACTOR HUM1 <=> (*MTRANS*) MOBJECT
    (*CONCEPTS* CONCERNING CNTRCT1) TO (*?*)) TIME G1380)
   applying INFER-FROM-TOPIC to CNTRCT1
      inferring undifferentiated political meeting
The inferred question concept is:
((<=> (sM-MEETING ACTOR HUM1 OTHERS (*?*) TOPIC CNTRCT1))
   TIME G1380)

searching memory for question concept
  searching directly for sM-MEETING
   searching for $MEET
   searching for $CONSULT
   searching for $PUB-REL-MEET
   ...

Begin, Sadat, Dayan, Hussein, and Assad at diplomatic
meetings, Carter during consultations, and a group of
Jewish leaders at a public relations meeting.
```

The only specification in the question which suggests an event context is thr Camp David Accords (CNTRCT1), an international contract. Characteristic events in which international contracts occur as topics are political meetings (sM-MEETING to CYRUS). Thus, CYRUS applies the strategy Find-from-Topic to infer that Vance would have discussed the Camp David Accords during political meetings. It then searches for political meetings. Because there are a number of different types of political meetings it knows about, CYRUS traverses memory individually for each of those. The context-instantiation and search strategies which guide that search will be discussed later.

Consider the following question:

(Q3-2) Have you talked to Begin recently about Arab-Israeli peace?

Because Arab-Israeli peace is an international issue, it has the context "political meetings" associated with it. Because Begin is a foreign diplomat, he has the context "diplomatic meetings" associated with him. "Diplomatic meetings" is a specialization of "political meetings". The more specific context should be searched for in a case like this.

3.2.3 Using locations for context construction

Location specifications can also be used by context-instantiation rules to infer event contexts. In answering "Did you see Mr. Smith at the pool yesterday?", a context associated with "pools" as locations would be appropriate for searching memory. "Swimming at the gym" or "going to the swim club" could be inferred and searched for. Similarly, in answering "When was the last time you were at a ski resort?", the location specification "ski resort", which has "ski trips" associated with it, can be used to construct and search for a "ski trip" context.

The context-instantiation strategy which allows for processing of this sort is Infer-from-Location. Using this strategy, if the location specified in a question has event contexts associated with it, those event contexts can be inferred and searched for.

Infer-from-Location

If locations or classes of locations specified have associated event contexts, predict the characteristic events associated with each of them that could also include other already-specified components.

Figure 3-3

In CYRUS, this strategy is used in answering questions such as "Were you in Europe last week?" and "Have you been to Asia recently?" CYRUS knows that events associated with foreign places are "diplomatic trips". Thus, in answering these questions, CYRUS predicts that it should look for diplomatic trips.

```
Enter next question:

>When was the last time you were in Egypt?

The question is:
((ACTOR HUM1 IS (*LOC* VAL POL6)) TIME TIME0)
The question type is "time"
The question concept is:
((ACTOR HUM1 IS (*LOC* VAL POL6)))
   applying INFER-FROM-LOCATION to POL6
     inferring diplomatic trip context
The inferred question concept is:
((<=> (sM-VIPVISIT ACTOR HUM1 DESTINATION POL6)) TIME TIME0)
searching memory for question concept
  searching directly for sM-VIPVISIT
 ...

 on DEC 10 1978.
```

In answering "When was the last time you were in Egypt?", CYRUS uses the facts that Egypt is a foreign country and that Vance would have been there as part of his job, to infer that he would have gone there on a diplomatic trip. It thus uses Infer-from-Location to infer a diplomatic trip to Egypt as the search context.

3.2.4 Context construction and time

Time specifications can also be used to infer event contexts. Consider, for example, the following two questions:

(Q3-3) While you were a lawyer, what did you usually discuss with your partners?

(Q3-4) What kind of exercise did you get last winter?

The first question can be answered by referring to event contexts characteristic of lawyers which would include partners in the law firm. The second can be answered by inferring winter sports contexts. The following context instantiation rule guides construction of event contexts from time specifications.

Infer-from-Time-Specification

If the time specification refers to a time period with characteristic events (an era) or a long term role with characteristic activities associated with it, predict those characteristic events of the time specification that could include other already-specified components.

Figure 3-4

The time period necessary to answer (Q3-3) above is an era (Kolodner, 1978, Schank and Kolodner, 1979). An era is a time span characterized by some long term role theme (Schank and Abelson, 1977) or long term state. Thus, eras can coincide with job situations, family roles, homeownership and school, to name a few. The most important piece of information an era holds is the role or state which characterizes it. If that role or state has event contexts associated with it, then they can be instantiated when an era is specified. Sports contexts associated with high school, for example, include pep rallies and football games. The quesion "While you were in high school, did you ever meet a football player?" might be answered by recalling "pep rallies", a kind of situation which happened during high school which involved football players.

3.2.5 Context construction and personal relationships

Another aspect of the target concept which can be used to infer an event context is the relationship between the respondent and others involved in a situation he is attempting to retrieve. This is the case when we ask CYRUS the question "Has your wife ever met Mrs. Begin?" There are many types of situations which Vance associates with his wife. Applying the strategy Infer-from-Participants would not be helpful in the case of his wife being the only other known participant. Using the relationship of Vance to his wife, however, the types of situations in which his family group and others would be involved can be inferred. Those situations are social situations. Applying "Infer-from-Participants" to the family group specification, then, CYRUS can infer that the occasion must have been a social occasion. The inference that this was a social occasion which included both the Vance's and Mrs. Begin allows the inference that Mr. Begin was also a participant in the event. "Infer-from-Participants" can be applied again to the occupational group set up by Vance and Begin's relationship to infer that the occasion was probably political. The following output from CYRUS shows how it applies these rules to answer this question.

```
Enter next question:

>Has your wife ever met Mrs Begin?

The question is:
((ACTOR HUM4 IS (*PROX* VAL G1571)) TIME G1572 MODE (*?*))
The question type is "verification"
The question concept is:
((ACTOR HUM4 IS (*PROX* VAL G1571)) TIME G1572)
   applying INFER-FROM-PARTICIPANTS
         to family group (HUM1 HUM4)
      inferring social occasion
   applying INFER-FROM-PARTICIPANTS
         to occupation group (HUM1 HUM60)
      inferring political occasion
    inferring political social occasion
The inferred question concept is:
((<=> (sM-SOCIAL-OBL ACTOR HUM1 OTHERS (HUM4 HUM60 G1571)))
   TIME G1572)
searching memory for question concept
  searching directly sM-SOCIAL-OBL
   searching for $PARTY
   searching for $DINE
   searching for $STATE-DINNER
   ...

 yes, most recently at a state dinner in Israel.
```

Using the relationships of the people, CYRUS answers this question by inferring that if it knew that Mrs. Vance and Mrs. Begin had met, then Vance would also have been present. It applies Infer-from-Participants twice to infer that it would have been a political social occasion. The political social occasions CYRUS knows about (sM-SOCIAL-OBL) are entertaining officials privately at parties and dinner, and state dinners. It searches for instances of each of those involving both couples, and finds a state dinner in Israel.

3.2.6 Implications of context construction

Notice that in using these strategies, there is always the chance that something relevant will not be retrieved because its context was not inferred. Vance's discussions about the Camp David Accords outside of political meetings are not recalled by CYRUS in answering "Who have you talked to about the Camp David Accords?" He may have given a speech about the situation or have talked to somebody about the accords at some social political situation (such as a state dinner). Similarly, if Vance's wife had met Mrs. Begin in some situation other than a social activity when the Vance's and Begin's were present, then retrieval would have failed.

Construction of contexts for retrieval is dependent on the information or cues available at the time of construction. A context can be chosen for construction only if some aspect of the retrieval specification refers to that context. In answering "Who is the most famous person you've ever met?", a person might fail to remember meeting a famous person in a museum because there is no available cue to initiate retrieval of museum experiences. On the other hand, if the person were prompted with "how about in a museum?", there would be a better chance of his retrieving that experience.

In judging the advantages and disadvantages of traversal/elaboration in the last chapter, a similar problem was mentioned. As in that case, the advantages of directed context construction seem to outweigh its disadvantage. The major advantage of context construction for the retrieval process is that it constrains search only to relevant contexts. Only relevant E-MOPs are ever searched.

3.3 Component-Instantiation strategies

(Q3-5) Who have you met with about the Camp David Accords?

(Q3-6) Why did you meet with Begin last July?

In the last chapter, we explained that during the traversal/elaboration process it is often necessary to elaborate on a target specification to enable retrieval. In the first question above, the question requires enumeration of similar episodes. Elaboration is needed to hypothesize differentiating features of those episodes so that individual instances can be retrieved. To answer the second question, elaboration might be needed to transform the time specification into some hypothetical related feature that is indexed.

Component-Instantiation strategies are the reasoning processes that are used to elaborate on concepts targeted for retrieval. These strategies get their guidance and information from two different places -- from an E-MOP's specifications, or from generally known relationships between particular components.

Consider, for example, specifications on the "negotiations" MOP:

"negotiations"

```
content frame:  participants = representatives of countries
                               involved in contract
                topic = contract of international importance
                duration = months to years
```

Figure 3-5

This E-MOP specifies that its participants are generally representatives of the countries involved in the contract under negotiation. Thus, if the target event were "negotiations about SALT", the participants in the event could be found by consulting the "negotiations" MOP to find the relationships between the different "negotiations" roles. In this way, the target event could be elaborated to be "negotiations about SALT with Gromyko" or "negotiations about SALT with Brezhnev" using E-MOP information and specific information from the target event.

Instead of MOP-specific information, component-instantiation can use very general information *about the usual relationships between* different kinds of event components. Included in the information we have about people, for example, are their places of residence, interests, roles, and occupations. The general information that people often do things in their place of residence can be used by component-instantiation strategies to infer the location for a particular event if the appropriate E-MOP cannot help in designating a place.

Component-instantiation is not an all or none process. Sometimes both MOP-specific and general information are needed for elaboration. An E-MOP may hold only a partial specification of a component to be filled out or a specification that cannot be filled in using other known event specifications. Consider, for example, a hypothetical "ski trip" MOP:

"ski trips"

```
content frame:  location = mountains,
                           ski resorts in the mountains
                participants = ...
```

Figure 3-6

Suppose we wanted to retrieve particular "ski trips" from this E-MOP. Its default setting is "in mountains" or "resort areas in the mountains". That specification does not give particular settings but gives enough information to apply other strategies to further specify the setting. Starting from that partial specification, mountainous areas or resorts in the mountains could be enumerated using other strategies. Although an E-MOP might not specify particular values for its components, it can put constraints on other instantiation strategies by providing partial knowledge.

When E-MOPs do not specify particular values for slot fillers, instantiation strategies use more general information about other specified components. Participants and places in episodes, for example, are normally related through the place being the residence or place of occupation of the participant, or a place where he can often be found. Thus, to answer a question such as "Where did you see Joe yesterday?", the places associated with Joe can be enumerated and checked without doing other memory search. It is component instantiation strategies which access those correlations.

Although component instantiation can get its information from two different places, the partial specifications provided by an E-MOP's content frame should always be used first. In that way, their constraints can guide application of other component instantiation rules. In other words, domain-related (i.e., MOP specific) specifications should always be applied during component instantiation before weaker domain-independent correlations.

Component instantiation strategies are associated in memory with the types of components they elaborate on. Strategies for participants, then, would be stored in a node in memory describing "participants". That node would also hold "Infer-from-Participants", the component-to-context instantiation rule "participants" has associated with it.

Strategies for inferring locations, times, and participants will be discussed below. It will always be assumed that MOP-specific correlations were taken into consideration before using the general correlations presented below.

3.3.1 Instantiating locations

Following are some of the rules for inferring place. Implicit in all of them is that the place inferred must not conflict with any partial specification of place (e.g., from an E-MOP's partial specification), that it must not conflict with other specifications in the Target Concept, and that all retrieval is for the time specified in the Target Concept.

1. Location can be inferred from important people participating in the Target Concept. Infer place from person specifications as follows:

   (a) If the location to be retrieved is related to occupation, and if the Target Concept refers to a person's occupation, retrieve his place of occupation.

   (b) Otherwise, use his place of residence, places he habituates, or nationality to the level of detail specified in the partial specification.

If, for example, this rule is applied when it is already known that the location should be a country, retrieve the actor's country of residence. If the location should be a city, his city of residence would be appropriate, and if a building, his house or some other appropriate type of building. As in the rule above, implicit in all of the following rules is that the specification should be to the level of detail implied in the partial specification in the target event.

2. Location can be inferred from organizations appearing in the Target Concept as follows:

   (a) Retrieve the organization's place of residence (i.e., its location).

   (b) Retrieve the place of the organization's affiliation (e.g., UNICEF is affiliated with the UN and can be expected to sometimes have its meetings there).

   (c) If an important representative of the organization is a participant, then retrieve his place of residence in the organization or places he habituates in the organization (e.g., his office or conference room).

   (d) If the partial location description specifies that the place is not the place of residence of the organization, retrieve places habituated by the organization in the locale of the Target Concept or places habituated by the important representative of the organization in the locale of the Target Concept.

3. Infer location from groups participating in the Target Concept as follows:

   (a) If the group is an ad-hoc group, infer place by inferring place for important members of the group.

   (b) If the group is affiliated with an organization, infer place by using rules for inferring place from rules for organizations (rule 3).

   (c) If the group is affiliated with a polity or locale, retrieve the polity or locale it is affiliated with.

   (d) Retrieve the place of residence of the group, its habitual meeting place, or its nationality.

4. Infer locations from locations by retrieving subparts and surrounding areas appropriate to the event class. (e.g., for sightseeing, retrieve famous subparts, for trips, retrieve surrounding country or locale)

5. To infer location from international contracts, use the sides of the contract, their capital cities, or a normal neutral location for dealing with international contracts (e.g., Belgium or Switzerland).

CYRUS uses location information associated with people, places, contracts, groups, and organizations in order to infer possible locations for events. In answering the question, "Who have you discussed SALT with?", for example, CYRUS first uses the context-instantiation strategy Infer-from-Topic to infer a context of a political meeting. Because the locations of political events are often foreign countries, and because political meetings are political events, CYRUS can make use of the sides of the SALT contract to infer that a meeting about SALT may have taken place in the Soviet Union or the USA. It thus elaborates the meeting context it is looking for and searches for political meetings in both the Soviet Union and the United States.

```
Enter next question

>Who have you discussed SALT with?

The question is:
((ACTOR HUM1 <=> (*MTRANS*) MOBJECT
       (*CONCEPTS* CONCERNING CNTRCT2) TO (*?*)))
The question type is "concept completion"
The question concept is:
((ACTOR HUM1 <=> (*MTRANS*) MOBJECT
       (*CONCEPTS* CONCERNING CNTRCT2) TO (*?*)))
   applying INFER-FROM-TOPIC to CNTRCT2
     inferring political meeting context
The inferred question concept is:
((<=> (sM-MEETING ACTOR HUM1 OTHERS (*?*) TOPIC CNTRCT2)))
searching memory for question concept
    additional information needed
applying strategies to elaborate question concept
  checking locative information on question concept
    sM-MEETING could have occured in USSR or USA
      searching for sM-MEETINGs in the USSR
       ...
      searching for sM-MEETINGs in the USA
       ...

Gromyko and Brezhnev at diplomatic meetings in the USSR,
and Carter and Brzezinski in consultations in Washington.
```

3.3.2 Inferring participants

Participants in an event include people, organizations, groups, and countries which take part in the event. When an E-MOP cannot specify the particular participants in an event, they can be inferred from locations, topics, and other participants. If we know, for example, that Vance attended a diplomatic meeting in Israel, then we can predict that one of the participants was probably Begin, and that other high-ranking representatives of the Israeli government, such as the foreign minister or defense minister might also have been present. We can make that prediction because we are aware of a correlation between topics and participants in political events. The following are rules for specifying participants.

1. Infer participating people by retrieving representatives of specified organizations, members of known groups, representatives of known countries, or person's associated with known organizations, groups, or countries fitting the initial description.

In answering a question such as "Have you met with NATO recently?", important representatives of NATO can be enumerated as meeting participants and meetings with each can be searched for.

2. Infer participating organizations by retrieving organizations known people or groups are affiliated with, and organizations affiliated with known countries.

3. Infer participating groups by retrieving groups known people belong to, and groups affiliated with known organizations and countries.

4. Infer participating countries by retrieving nationalities, affiliations, and places of residence of known people, groups, and organizations, or by retrieving sides of a known contract or issue.

CYRUS uses some of these rules to better specify participants in events. The E-MOP "political meetings" (sM-MEETING in CYRUS) specifies that its participants be people or organizations representing the sides of a contract being discussed or negotiated. Thus, in answering a question such as "Who have you discussed the Camp David Accords with?", after applying the context-instantiation strategy Infer-from-Topic to infer a political meeting context, CYRUS can consult its knowledge about the Camp David Accords to find out who it involves. After discovering that it involves Israel, Egypt, and the United States, CYRUS can use Component-Instantiation strategies to infer that the participants in meetings would be representatives of those countries. CYRUS can further specify possible participants by recalling important diplomats of those three countries and checking whether Vance had met with any of them about the Camp David Accords.

```
Enter next question:

>Who have you discussed the Camp David Accords with?

The question is:
((ACTOR HUM1 <=> (*MTRANS*) MOBJECTS
    (*CONCEPTS* CONCERNING CNTRCT1) TO (*?*)))
The question type is "concept completion"
The question concept is:
((ACTOR HUM1 <=> (*MTRANS*) MOBJECTS
    (*CONCEPTS* CONCERNING CNTRCT1) TO (*?*)))
  applying INFER-FROM-TOPIC to CNTRCT1
    inferring political meeting context
The inferred question concept is:
((<=> (sM-MEETING ACTOR HUM1 OTHERS (*?*) TOPIC CNTRCT1)))
searching for question concept
  searching directly for sM-MEETING
    additional information needed

applying strategies to elaborate question concept
   ...
 elaborating OTHERS
     OTHERS are RT-DIPLOMAT of Israel, Egypt, or the USA
   searching for sM-MEETING with OTHERS RT-DIPLOMAT of
Israel
      additional information needed
      elaborating OTHERS
         OTHERS could be Begin, Dayan, or Weizmann
        searching for sM-MEETING with OTHERS Begin
          ...
        searching for sM-MEETING with OTHERS Dayan
          ...
        searching for sM-MEETING with OTHERS Weizmann
          ...
    searching for sM-MEETING with OTHERS RT-DIPLOMAT of
Egypt
       additional information needed
       elaborating OTHERS
          OTHERS could be Sadat
         searching for sM-MEETING with OTHERS Sadat
           ...
      ...

Israeli and Egyptian diplomats including Begin, Dayan,
Weizmann, Sadat, and Khalil, and American diplomats
including Carter and Brown.
```

3.3.3 Inferring time

Time cues can often be provided by generalized E-MOP information. Many activities are specific to a particular time of the year (e.g., sledding in the winter, final exams in the early winter or spring), while some are specific to a particular time of one's life (eras). Exams, for example, are particular to times when one is a student. For Cyrus Vance, meetings with heads of state are probably particular to times when he has had a diplomatic role. Each E-MOP includes on its content frame temporal information, i.e., generalizations about when it usually happens.

In addition, some objects, people, and places have temporal information associated with them. Boots and heavy coats are winter clothing and an event involving either of them is not likely to have occurred in the summer. Similarly, experiences with a person one knows only through summer camp can be inferred to have happened during the summer and at a time of life which included going to summer camp. Experiences with a person one associated with only in college would have happened during college years.

Time information inferred from objects, people, and places can be found directly associated with the object or person, or can be inferred from characteristic activities associated with the object or person. If a person who has gone to the Caribbean during the winter is asked about an experience that would have happened in the Caribbean, he will be able to attach time information to the experience by retrieving the default time from a characteristic Caribbean activity, namely vacationing there.

Time representations can refer to particular times or can be specified in terms of events, characteristic events, role theme , (Schank and Abelson, 1977), or eras. An era (Kolodner,1978) is a time span in a person's life characterized by some long term role he is playing. Eras are usually associated with occupational, family, and social roles. Thus, for Vance, the time period when he worked as a lawyer was an occupational era characterized by his being a lawyer. Eras can have associated with them the characteristic roles and activities a person is involved in during that time span, as well as other eras going on at the same time and other eras it is related to in time. An educational era comprising going to professional school must come after an era of going to college. Thus, they can be helpful in better specifying the time for an event.

The specification, "while I was a lawyer" has an occupational era associated with it which can be consulted to infer better time specifications. "When I was a teenager" can be expanded upon to include eras going on at that time (such as high school) and events characteristic of that time which might serve as time specifications on events in memory.

Time cues, then, can be provided by E-MOPs, people, objects, places, and partially specified times. Similarly to the way locational specifications are inferred, time specifications should first be retrieved from content frames of the relevant E-MOPs. The constraints posed by those specifications can be used to retrieve time information

from other event components. Also, as the example above shows, time specifications themselves can be used to expand on time cues. Following are some rules for inferring time.

1. Infer time cues by retrieving times associated with already-specified components, such as people, organizations, and countries.

2. Infer time cues by retrieving times associated with characteristic events associated with already-specified components.

3.4 Enumeration strategies and maps

Q: Which of your colleagues at work have you had lunch with?

A: There's the regular lunch crew, I've had lunch with all of them. Then, there's Smith at the end of the hall, I've never had lunch with him. Next to his office is Jones. We had lunch the other day. In the office across from there is ...

Often component instantiation requires enumeration of possibilities. How is that possible if memory is not organized in lists? One way we've observed people doing enumeration is by using some well-known map or organizational chart. In the example above, we imagine the person reconstructing and going through the floor plan of his office building to enumerate all the employees he might have lunched with. Recall that in answering "What museums have you been to?", some people seemed to recall museums by mentally walking through a map of a familiar area, stopping at each important place and recalling if they had been to a museum in that place. Thus, their answers were something like "Let me see, have I been to a museum in Rome, in Naples, in Florence, ..." The set of component instantiation strategies which allow enumeration of this sort are called _enumeration strategies_.

Enumeration strategies allow cues to be enumerated in an orderly fashion. They work by sorting through well-known or easily-constructed maps and charts to compute sets of possibilities for particular components of the target event. To enumerate trips to Europe, for example, it would be appropriate to walk through a map of Europe checking at each important point to see whether there had been a trip including that place. Similarly, to enumerate the players on a particular football team, we might expect a football fan to picture the playing field and go through each position on the field picking out the people on the team who play that position. Enumeration strategies, then, can be used for extensive search of both generalized and episodic memory, or for keeping track of a number of choices.

Any well-known map or chart can be used for enumeration. Imagining a map of the United States is the only way some people can name all 50 states. Similarly, well-known charts of organizations or objects are convenient for enumeration. A floor-plan of offices might be convenient to use in enumerating the people who work in a building. A chart of organizational set-up can be used to name all the directors of a large organization. To remember all the doctors a person has been to, a chart of the human body could be used, and doctors specializing in each particular bodypart could be enumerated.

One place where it would be appropriate for CYRUS to apply maps is in elaborating on a place specification. Often, CYRUS is asked questions about the Middle East, Europe, or Asia. In retrieving episodes that happened in those places, it is often convenient to specify each of the countries that could have been involved. Thus, in recalling whether Vance was in the Middle East last November, after using context-instantiation to infer a diplomatic trip context, one of the things CYRUS does is to attempt elaboration of the trip description. In elaborating the destination of the diplomatic trip (sM-VIPVISIT), CYRUS enumerates the countries it knows to be in the Middle East (LOC17). It thus attempts to recall trips to Israel, Egypt, Jordan, and Syria, and responds as follows:

```
Enter next question:

>Were you in the Mid East last November?

The question is:
((ACTOR HUM1 IS (*LOC* VAL LOC17)) MODE (*?*) TIME G0832)
The question type is "verification"
The question concept is:
((ACTOR HUM1 IS (*LOC* VAL LOC17)) TIME G0832)
   applying INFER-FROM-LOCATION to LOC17
     inferring a diplomatic trip context
The inferred question concept is:
((<=> (sM-VIPVISIT ACTOR HUM1 DESTINATION LOC17)) TIME
G0832)
searching memory for question concept
  searching directly for sM-VIPVISIT
    additional information needed

applying strategies to elaborate question concept
  checking locative information on question concept
    applying map of LOC17 to specify place
      sM-VIPVISIT could have been to Egypt,
           Israel, Syria, Jordan, or Saudi Arabia
     searching for diplomatic trips to Egypt
       ...
     searching for diplomatic trips to Israel
       ...
  ...

 No.
```

Although maps and charts are used by people for enumeration, CYRUS does not have an extensive variety of maps. The ones it has are implemented as short lists. The reason for this lack in CYRUS is that the problem of organizing a map or chart in memory has not been addressed. Many of the maps and charts people seem to use are visual and easy to reconstruct through visual cues. Without a good handle on what a visual cue is, however, their implementation would be incomplete and probably list-like, at best.

3.5 Instantiating events from events

The last set of instantiation strategies are context-to-context instantiation strategies -- strategies for constructing event contexts related to specified events. These strategies are used both for elaboration and during alternate-context search. During alternate-context search, search strategies guide application of these context instantiation strategies. To retrieve meetings from CYRUS, for example, it is often necessary to search for related trips and conferences. It is context instantiation strategies which construct contexts for those related events.

Context instantiation strategies use specifications on E-MOPs of how they relate to other E-MOPs to guide their construction. "Diplomatic meetings", for example, are usually included in "negotiations". The topic of the negotiations includes the meeting topic, the participants in the negotiations include the meeting participants plus other representatives of the contract sides. Using these specifications, the "negotiations" episode any particular diplomatic meeting was part of can be inferred. A "diplomatic meeting with Gromyko about SALT", for example, must have been part of "SALT negotiations including Gromyko and other Russian and American diplomats".

As with component-instantiation strategies, context instantiation rules use both MOP-specific information and general information about containment, causality, and time relationships between events. The information above about the relationship between "diplomatic meetings" and "negotiations" is an example of some of the MOP-specific information these strategies use. An example of the general information they use is that "events contained in other events have time and place which are contained in the time and place of the larger event".

Context instantiation strategies have four steps. They are listed below:

Context-to-Context Instantiation Strategies

1. identify the most specific E-MOPs the target event fits into

2. retrieve the requested content frame property from those E-MOPs

3. transfer specifications on the target event to the new context being instantiated making use of specifically-defined relationships and default content frame relationships

4. apply component instantiation strategies to further specify underspecified role fillers

Figure 3-7

The most specific E-MOPs a target concept can fit into are those which have no indices that can be traversed using target event specifications. Specific E-MOPs for a meeting between Vance and Begin about Egyptian-Israeli peace might be "meetings with Begin", "meetings with Israelis about Egyptian-Israeli peace", "meetings about Egyptian-Israeli peace", "meetings about Middle East peace", etc.

Once a specific E-MOP has been found, the requested content frame property is retrieved from it. Recall that these strategies are called either by the elaboration process or by search strategies. Both request construction of a particular content frame property. These properties correspond to the ways events can relate to each other -- through containment, various kinds of causality, preceding, following, etc. An E-MOP's specification of a related E-MOP will specify usual relationships between its role fillers and those of the context being constructed. "Negotiations", for example, specifies that the topic of a "diplomatic meeting" is a subtopic of the negotiations topic, and that participants are representatives of the countries involved in the negotiations.

In the third step, relevant specifications are transferred from the target event to the context being instantiated. Thus, if a particular negotiations episode is about SALT and with the Russians, the meeting constructed from it will be about some aspect of SALT and be with a Russian representative.

At this point, general information about the specified event relationship is used. When there are no specifications for instantiating some component of the new context, default information about the type of relationship between the target event and the event being constructed is used. In particular, time, place, and whole/subpart inferences can be made based on the relationship the

events have. The time of an event contained in the target episode, for example, is during the time specified for the target event. The time for an enabling event is before the time specified by the target event. Participants of events in an episode contained in the target event include some of its participants. The location of an episode contained within the target event is some location within or near the target event.

The fourth step in instantiating a new context is to apply component instantiation rules to better specify components. Already-specified constraints help to narrow the work of the component instantiation strategies.

There is a context instantiation rule corresponding to each type of relationship events have to other events. Events can be contained in episodes, contain other events themselves, be before after, or during other episodes. Events are also causally related through enablement or direct causality. Events that cause or enable other events come before them in time. Events caused or enabled by other events come after them. Each of these relationships has a context instantiation rule associated with it.

The first and fourth steps of each strategy are alike. The second and third depend on the relationship between the target event and the context being instantiated.

The strategy "Instantiate-Seq-of-Events", for example, constructs aspects of the sequence of events of a given episode. It takes as input an episode description (the target event), and optionally a sketchy description of the event to be instantiated. The default sequence of events is retrieved from the E-MOP the target event fits into, and each appropriate event in the sequence is instantiated. Because events of a sequence of events are part of a larger specified episode, time specifications of events within the episode will be within the time specifications of the episode, place specifications will be within the place specified on the given episode, and participants will include a subset of the people involved in the entire episode.

Applying this strategy to a diplomatic trip to Israel for the purpose of negotiating Arab-Israeli relations, and requesting instantiation of the trip's entire plausible sequence of events, would produce a list of events including flying to Israel, being welcomed at Ben Gurion Airport by a high Israeli official, meeting with Begin and Israeli officials at some unspecified place within Israel (probably in Jerusalem or Tel-Aviv), attending a state dinner hosted by Begin, and staying over at a diplomatic residence in Tel Aviv or Jerusalem. Applying this strategy to the same diplomatic trip and specifying that it should instantiate a diplomatic meeting would produce a diplomatic meeting with Israeli officials in Israel. The rules for Instantiate-Seq-of-Events follow.

Instantiate-Seq-of-Events

(1) Identify the E-MOP the specified event fits into.

(2) Get the default sequence of events for that E-MOP.

(3) Go through the event sequence one by one transferring components as specified and making the following inferences when more specific information is not available on the E-MOP.

   (a) Time specifications are within the time specified on the input event.

   (b) Place specifications are within the place specified on the input event.

   (c) Participants in any of the events being instantiated include a subset of the known participants plus a number of as yet unspecified participants.

(4) Use relevant Component-Instantiation strategies to further specify components of each event, using the already-instantiated partial descriptions of each component as constraints.

Figure 3-8

This strategy infers as much as it can from MOP-specific and target event-specific information (in step 3) to fill in details of the events it is instantiating. It then uses knowledge about the relationship between the events to better specify the event being constructed (steps 3 a, b, and c). Only after applying that domain-specific and event-relationship-specific information does it call component-instantiation strategies to further specify particular components of the new events.

Other context-instantiation strategies work similarly. Each specifies in step 3 the relationships between the components of the input context and the components of the context being created. Following is a list of all the context-to-context instantiation strategies. For details of each, see appendix A.

Context-Instantiation Strategies

Instantiate-Enablements
Instantiate-Preconditions
Instantiate-Results
Instantiate-Reasons
Instantiate-Enabled-Events
Instantiate-Larger-Episodes
Instantiate-Seq-of-Events
Instantiate-Preceding-Events
Instantiate-Following-Events
Instantiate-Standardizations

Figure 3-9

Context instantiation from E-MOP information serves two purposes. In searching memory, it is often necessary to search for alternate contexts. One purpose of instantiation strategies is to construct those contexts for search. During retrieval, these strategies are guided by Search strategies, which will be explained in the next chapter.

The other purpose of context instantiation strategies is in reconstructing probable contextual details when they are not specified, but when they are wanted. If E-MOPs hold information about the kinds of episodes they are related to, that information can be used to infer "what must have happened", regardless of whether "what must have happened" can actually be found in memory.

In answering the question, "How did Vance become Secretary of State?", for example, the strategy "Instantiate-Enablements" can be used to instantiate the normal enablement conditions for becoming secretary of state if no specific enablement conditions are known. Thus, the question can be answered by "He was appointed by Carter."

3.6 Representations of events in E-MOPs

What can instantiation processes tell us about the representations of events in memory? Instantiation processes are used during retrieval to infer aspects of a target event's context. They can also be used to elaborate on specifications stored in memory. Thus, an important implication of the ability to make the inferences made by instantiation strategies is that _all details of events need not explicitly be stored_. If generalized knowledge is available, it can be used to infer normative details when needed.

If constructive processes can be used to reconstruct unspecified details of events, then the question arises as to how much detail is necessary within memory to refer to events stored there. Up to now, the

assumption has been that E-MOPs organize pointers to actual events, and that events in memory point to the other events they were related to. CYRUS' memory has this organization. In terms of a computer model, this organization has the advantage that large chunks of episodes can always be retrieved in detail without mistakes. Is all of that direct reference necessary?

References to actual events can be costly in terms of storage. Since E-MOPs store the similarities between the events they organize, and since the apparatus for reconstruction from partial descriptions is available in memory, references to complete events are overly redundent. They repeat normative information found at the E-MOP level. All we really need to store are those aspects of an event which are different than, or add to, an E-MOP's specifications. Merging those differences and the E-MOP's norms during retrieval produces a specification of an entire event. References to events in E-MOPs need only enough information to enable that reconstructive process to happen.

Consider, for example, how the following event could be stored in a "diplomatic meetings" MOP.

EV3-1: Vance went on a diplomatic trip to China to discuss Chinese-American relations with Chinese leaders.

If that E-MOP held the information that diplomatic trips are to foreign countries, usually for negotiations, and include diplomatic activities, then this particular trip could be stored in that E-MOP as follows:

```
                        "diplomatic trips"

content frame:
     Vance is a participant
     participants include state department aides
     origin is Washington
     destination is a foreign country
     duration is a few days
     sequence of events is:
         fly + welcome + diplomatic activities + fly
     purpose is negotiations
differences:             |             \
        /                |              \
       /                 |               \
  destination      purpose is      sequence of events
  is China         diplomacy       includes meeting
     /                   |          about Ch-Amer rel
    /                    |               \
included meeting    destination         purpose was
 about Ch-Amer rel   was China           diplomacy
purpose was         included meeting    destination
 diplomacy           about Ch-Amer rel   was China
```

Figure 3-10

Its differences and additions to the norms include the features that (1) it was to China, (2) its purpose was diplomacy (as opposed to negotiations), and (3) it included at least one meeting about Chinese-American relations in its sequence of events. Thus, those three features are features that the episode is indexed on. The descriptions of the event at each of those index points, however, need hold only the features that differ from the E-MOP's norms and that are not part of their index. They do not need to hold an entire event description. Thus, the event is described by the features "included a meeting about Chinese- American relations" and "purpose was diplomacy" when indexed under "destination is China". Using those features plus the norms for "diplomatic trips", there is enough information to reconstruct the trip as follows:

> Vance went on a diplomatic trip to China for diplomatic reasons. The trip included a meeting about Chinese-American relations. Its first event was a flight to China; there was a welcome at the airport there; there were a number of diplomatic activities, one of which was a meeting about Chinese American relations; and there was a flight back to the United States."

Events in memory refer to other episodes they are related to. How much detail must those references include? If events stored in E-MOPs must include only enough information to enable their reconstruction, then it follows that references to episodes a particular event is related to should need to specify only those features that are necessary for its reconstruction. A reference to a meeting that happened during EV3-1, for example, should need to specify no more than the fact that it was a meeting, its topic was Chinese-American relations, and its participants were Chinese leaders. Using that information, plus generalized information about "diplomatic meetings" and information relating "diplomatic trips" to "diplomatic meetings", a more complete description of the meeting, including its place and location, could be inferred.

A meeting that happened during EV3-1, however, is itself stored in a "diplomatic meetings" MOP. All of its meeting-relevant features are recorded in that E-MOP. That information, then, does not need to be part of the trip specification. Instead, all that is needed is enough detail to recognize the meeting in the "diplomatic meetings" MOP. With that amount of detail, the meeting could be retrieved and reconstructed from the "diplomatic meetings" MOP if necessary. If, for example, that meeting were the only one about Chinese-American relations, then it would be sufficient for the trip to specify that it had as an embedded event a "diplomatic meeting about Chinese-American relations". The meeting could be recognized with that specification and no additional details of the meeting would need to be specified among the features of the trip to China.

References to events related to an episode, then, need contain only enough features to enable their retrieval. A more complete description can be obtained by retrieving and reconstructing other details of that description. This organization provides economy of storage by storing only differences from and additions to generalized E-MOP information at any index point. Similarities are stored only once at the E-MOP level.

3.7 Summary

Instantiation strategies are the strategies which are used for construction and elaboration of contexts for retrieval. Context-instantiation strategies are used to construct contexts for search, while component-instantiation strategies are used to elaborate on those contexts.

There are two kinds of context-instantiation strategies -- (1) those that are used to construct contexts for search from episodic information associated with question components, and (2) those that are used to construct episodic contexts from event specifications and their associated E-MOPs. The first set of these strategies, component-to-context instantiation strategies, is used in the first step of the retrieval process as a way of selecting relevant E-MOPs for traversal. The second set, context-to-context instantiation strategies, is used during elaboration and by search strategies.

Component instantiation strategies use information about the relationships between objects and people in order to predict event features which are not specified. Thus, if a "speech" MOP held the information that their topics normally correspond to the interests of the participants, the topic of a speech could be inferred if the participants were known.

CHAPTER 4

Event-Oriented Search Strategies

4.1 Introduction

Name all the amusement parks you've been to.

> There's an amusement park near my parent's house that we used to go to when we were kids -- Willow Grove. When I've gone to amusement parks since then, it's either been as part of an outing when a friend was in town or as part of a trip. Once, we went to an amusement park before going to a concert at the summer music school. That was with Howard and Lauri. Another time we went to an amusement park when John and Wendy were visiting, one around here someplace. I've been to Disney World twice, and to Disney Land last summer while we were in California. And I've been to Great Adventure once when I was in Philadelphia visiting my parents. I've also been to a bunch of amusement parks on the beach -- in Atlantic City, Coney Island, Santa Cruz, probably some others I can't remember ...

In searching memory extensively to remember an event, it is often helpful to search for episodes related to the target event. In trying to remember experiences at amusement parks, for example, the person above recalled friends visiting, trips around the United States, and going to amusement parks at the beach in order to recall his experiences. Alternate contexts people choose for search are not chosen randomly. The person above thought about the kinds of episodes that could have been related to amusement parks. He did not think about football games, conferences, or reading mystery books to remember amusement park experiences.

Search strategies work by calling instantiation strategies to construct contexts related to a target event, searching for those events, and then seaching the context of retrieved events for the target. If E-MOPs keep track of other E-MOPs they are related to, then that information can be used by search strategies to choose alternate

search contexts.

People seem to do extensive memory search under two circumstances: (1) when the event being searched for is obscure, and (2) when many events corresponding to a general specification must be found.

What is an "obscure event"? Consider the following question:

(Q4-1) Have you ever rented a Chevy?

Suppose a person who had rented cars many times were answering this question. People renting cars normally rent by size and not by manufacturer. Thus, unless a person once had a particularly bad or good experience renting a Chevrolet, he would probably not be able to retrieve an experience renting one without further specification. When he attempted retrieval, he might recall the following -- "I've rented cars many times, I need more information to find recall renting a Chevy". If he cannot elaborate on additional features of "Chevy renting", then he will have to answer "I don't know", although the event might be in memory. This is an example of an obscure event. It is one which might be present in memory, but for which there is not enough generalized information available to do the necessary elaboration to get it out.

When retrieval fails because of inability to produce a sufficient elaboration, search strategies can be applied to search for alternate contexts related to the event which might be more easily retrieved. In one case, we observed a person doing the following reasoning in order to recall if she had ever rented a Chevy:

> I rent cars during trips when I need to be able to travel easily. In California, it is impossible to get around without a car. On my trip there last summer, I rented a Fairmont, and on my last trip there, I had a Toyota. Neither of them was a Chevy. What kinds of bad experiences have I had with rental cars? Was one of those cars a Chevy? I once missed a plane because I couldn't find the rental car place to return my car -- that was a foreign car of some kind. Another time, I rented a car that was too big for me -- I couldn't see over the steering wheel. It was the only car they had available. That might have been a Chevy, I have no idea if it was or not.

In giving this answer, the respondent recalled trips during which she might have rented cars, and rental car experiences during those trips. She then recalled bad experiences while renting a car to see if any of those involved Chevrolets. In other words, she searched memory for episodes related to renting cars.

In many ways, searching for an obscure event is similar to searching for many instances of a very general event specification, i.e., searching for instances of an event which has happened many times. In recalling "diplomatic meetings about SALT", CYRUS receives the message from memory "there have been many meetings about SALT, I need more information to retrieve individual ones". Meetings about SALT are

AD-A093 084   YALE UNIV NEW HAVEN CT DEPT OF COMPUTER SCIENCE   F/G 9/2

RETRIEVAL AND ORGANIZATIONAL STRATEGIES IN CONCEPTUAL MEMORY: A--ETC(U)

NOV 80   J L KOLODNER   N00014-75-C-1111

UNCLASSIFIED   RR-187   NL

2 of 3

AD A093084

elaborated as much as possible. When CYRUS has done as much elaboration as it can and has not retrieved all meetings, it continues its search by searching for diplomatic trips and summit conferences that could have been related to the meetings. Their recall allows recall of additional meetings.

The major difference between these two reasons for alternate-context search is the certainty of knowing whether or not something will be found. There are definitely instances to be found of an episode which has happened many times and which has an E-MOP associated with it. There may or may not be episodes of an obscure event, one which seems to fit into an E-MOP in memory but which has no known E-MOP which corresponds to it.

Why should retrieval of related contexts aid retrieval of a target event? Any particular event that occurs happens in the context of other events going on around it, and thus must refer to those other events. A related context might be less common or better specified in its E-MOP than the target concept. In that case, it would be easier to retrieve. Once retrieved, a related context can provide pointers or references to events which correspond to the target.

4.2 Context search

Search strategies each have four steps. In the first step, the appropriate context-instantiation strategy is called. Next, retrieval of the newly constructed concept is attempted. If it is found, the third step, which involves searching its event context for the target event, is applied. If that fails, appropriate parts of the event context of the retrieved event are reconstructed and used to better specify the target concept, i.e., as a means of elaboration. Retrieval of the better-specified target can then be attempted. Those four steps are listed below:

Event-Oriented Search Strategies

1. call the appropriate context-instantiation strategy to construct a new context

2. traverse memory attempting to retrieve that context from memory

3. search appropriate aspects of the _episodic context_ of the concepts retrieved for the original target event

4. if that fails, call context instantiation strategies to reconstruct parts of the event context of concepts retrieved corresponding to the target. Attempt retrieval of the better-specified target concept again.

Figure 4-1

The _episodic context_ of an event includes all of the events and episodes which are causally and temporally related to the event. That includes events it contains in its sequence of events, episodes it is part of, states and events which caused and enabled it, states and events it caused and enabled, and events which preceded and followed it.

Search strat(;ies guide search of an event's episodic context by specifying which aspect of a retrieved alternate event should be identified. When searching for an episode by searching for the kinds of events it could contain, the episodes retrieved are checked to see if the events they were part of correspond to the target. For example, when searching for a "trip to Egypt" by recalling "sightseeing experiences at the pyramids", after retrieving the sightseeing episode, the trip it was part of should be referenced and checked to make sure it matches the target "trip to Egypt".

4.3 The strategies

Search strategies have three major steps -- (1) they call context-instantiation strategies to construct alternate contexts for search, (2) they attempt retrieval of those contexts, and (3) they guide search of those contexts for the original target concept. Search within the retrieved alternate episode depends on the relationship between the target concept and the retrieved alternate episode retrieved from memory. As for context-to-context instantiation strategies, there is one search strategy for each type of relationship between events. In the next sections, search strategies will be explained in detail.

4.3.1 Sequence of events

One event relationship useful in search is "sequence of events". Sometimes retrieval of an event that could have been contained in the target concept can aid search for the target. In order to enumerate his trips to Europe, for example, we observed one person who recalled significant sightseeing and museum experiences in Europe to remember particular trips he had made there. When CYRUS answers the question "What summit conferences have you been to recently?", it searches memory for diplomatic meetings and state dinners that could have been part of a summit conference. If it finds any, it retrieves the larger episodes they were embedded in to see if they were in fact summit conferences.

The strategy used in these examples is "Find-From-Seq-of-Events", defined below:

Find-From-Seq-of-Events

IF a possible sequence of events for the target concept can be inferred

THEN

(1) Instantiate each element of the sequence of events using the Instantiate-Seq-of-Events strategy, and fill in its slots as well as possible by applying Component-Instantiation strategies to the known components of the Target Concept.

(2) Search for each inferred concept in memory.

(3) If one is found, and if it refers to its larger episodes, see if one of them corresponds to the Target Concept.

(4) If it does not refer to its dominating episodes, apply Instantiate-Larger-Episodes to instantiate a larger episode corresponding to the target concept that it could have been part of, and search for that episode in memory.

Figure 4-2

This strategy directs application of the appropriate instantiation strategy (Instantiate-Seq-of-Events) to construct a hypothetical event-component of the target concept, and then attempts to retrieve that new context from memory. If retrieval is successful, the strategy directs search of the episodic context of the event found for the target

event. In this strategy, the target event is checked against larger episodes the event was part of. If there are no specified dominating episodes, Instantiation strategies are called to reconstruct the dominating episodes that "must have been going on", and those are checked against the target concept.

When we ask CYRUS to search memory for all of the summit conferences Vance has attended, one of the strategies it applies is Find-From-Seq-of-Events. *Using this strategy, CYRUS calls instantiation* strategies to construct contexts for diplomatic meetings with heads of state, addresses by heads of state, and state dinners (the kinds of meetings, speeches, and state dinners that could have happened during summit conferences), and searches for each of those. In the following example, we see CYRUS doing that.

```
Enter next question:

>How many Summit Conferences have you been to?

The question is:
((<=> (sM-SUMMIT-CONFERENCE ACTOR HUM1)) QUANTITY (*?*)
   TIME G0659)
The question type is "concept completion"
The question concept is:
((<=> (sM-SUMMIT-CONFERENCE ACTOR HUM1)) TIME G0659)
searching memory for question concept
  searching directly for sM-SUMMIT-CONFERENCE
   found (GN420)
applying strategies to search memory
  ...
  applying Find-From-Seq-of-Events to search for
       possible components of sM-SUMMIT-CONFERENCE
     searching for $WELCOME
     searching for $MEET
     searching for $SPEECH
     searching for $WORKING-MEAL
     searching for $STATE-DINNER
      found (GN475 GN526 GN577 GN514)

One about SALT and one about the Camp David Accords.
```

CYRUS first attempts to search for summit conferences that Vance has attended. Because he has attended many summit conferences, they are organized in an E-MOP which requires specification of indices for retrieval of individual events. By signaling the retrieval process that traversal has arrived at an E-MOP and not an event, memory in effect is telling retrieval that there are numerous summit conferences in memory and that search should continue. Operating without component instantiation strategies (one of CYRUS' modes of operation), CYRUS returns the most recent event in the summit conference MOP (most recent

events are stored specially in CYRUS), and continues search by applying search strategies to search for other contexts that might help it to find summit conferences.

One strategy CYRUS applies is "Find-From-Seq-of-Events". That strategy directs search for the kinds of activities that normally happen during summit conferences. Thus, CYRUS searches for welcomes by heads of state to conferences, meetings with heads of state about important topics, speeches by heads of state about important issues, and working meals and state dinners attended by heads of state. After finding events which match each specification given, it looks at the larger episodes each was a part of to see if they had happened during a summit. If so, it has retrieved the summit conferences it is looking for. CYRUS recalls two summit conferences in this way -- one about SALT and one about the Camp David Accords.

Similarly, if we ask CYRUS to recall all of Vance's trips to Egypt, it applies the strategy "Find-From-Seq-of-Events" while searching its memory, and searches for meetings he would have had there, speeches he might have given, flights, etc.

```
Enter next question:

>How many times have you gone to Egypt recently?

The question is:
((ACTOR HUM1 <=> (*PTRANS*) OBJECT HUM1 TO POL6) TIME G0742
      QUANTITY (*?*))
The question type is "concept completion"
The question concept is:
((ACTOR HUM1 <=> (*PTRANS*) OBJECT HUM1 TO POL6) TIME G0742)
     applying INFER-FROM-LOCATION to LOC6
       inferring diplomatic trip context
The inferred question concept is:
((<=> (sM-VIPVISIT ACTOR HUM1 DESTINATION LOC6)) TIME G0742)
searching memory for question concept
  searching directly sM-VIPVISIT
   found (GN528)
applying strategies to search memory
    ...
  applying Find-From-Seq-of-Events to search for
         possible components of sM-VIPVISIT
     searching for $FLY
     searching for $WELCOME
     searching for sM-VIP-TRIP-DO
     searching for sM-CONFERENCE
     searching for $MEET
     searching for $SPEECH
     searching for $REPORT
     searching for $FLY
      found (GN539 GN531)

 Four times in the past 6 months.
```

CYRUS first uses an instantiation strategy to infer that the context for being in Egypt must be a diplomatic trip. It then goes on to search directly for trips to Egypt. Because there are many of them, it continues by applying search strategies to aid its retrieval. The flights ($FLY) CYRUS looks for while searching for trips to Egypt are flights to and from Egypt, the welcomes are welcomes by Egyptian diplomats at the Cairo airport. The conferences it looks for are about Egyptian concerns, involve Egyptian diplomats and take place in Egypt, as do the diplomatic meetings ($MEET), speeches ($SPEECH), and press conferences ($REPORT) it searches for.

4.3.2 Using standardized methods for retrieval

Some E-MOPs don't have a standard sequence of events, but instead have more than one way of happening. For example, traveling can be done by air, by train, by car, etc. In order to find traveling episodes, then, it is reasonable to think of different types of travel, and to retrieve instances of each. A person answering the question "how many times have you been to the Carribean?" might try to think of times he had flown there and times he had gone there on a cruise. The strategy that directs retrieval of standard ways of doing an activity is called "Find-From-Standardizations".

Find-From-Standardizations

IF the Target Concept has a number of standard ways of being done

THEN

(1) Use Instantiate-Standardizations to instantiate each standard way of doing the Target Concept's activity that is appropriate to the rest of the target specification. Use Component-Instantiation strategies to fill slots in the hypothesized standardizations as well as possible.

(2) Search memory for instances of each of those standard ways.

(3) If one is found, it is a match.

Figure 4-3

This strategy hypothesizes appropriate standard ways of doing the target activity and searches for each of them in turn. Retrieval of one of the standardizations constitutes a match to the target concept. That means episodic context need not be searched once a standardized method is found. To retrieve trips to England, for example, a businessman might first try to remember business trips to England, and then try to remember vacations he took there. To remember diplomatic conferences, one type of conference CYRUS searches for is summit conferences.

We have observed that search strategies such as these are used extensively by people. Imagine how a person might attempt to remember trips he has taken. He might apply the strategy Find-From-Seq-of-Events to search for travel experiences. In order to search for travel, he may apply the strategy Find-from-standardizations and search individually for standard types of travel. Let's suppose he searched for air trips

and train trips, but did not search for cruises. Applying this strategy, he would fail to retrieve travel on ships, and thus might fail to retrieve some of the trips he had been on. Failure to apply a strategy completely can result in retrieval failure.

One of CYRUS' E-MOPs with standard methods is "political meetings" (sM-MEETING). Standard political meetings include "diplomatic meetings" ($MEET, meetings with foreign government officials), "consultations with other U.S. government officials ($CONFER), and "public relations meetings" ($PUB-REL-MEET). Thus, when CYRUS is searching for undifferentiated political meetings, it applies the strategy Find-from-Standardizations to search for instances of each type of meeting individually.

```
Enter next question:

>Who have you talked to about SALT?

The question is:
((ACTOR HUM1 <=> (*MTRANS*) MOBJECT
   (*CONCEPTS* CONCERNING CNTRCT2) TO (*?*)) TIME G1062)
The question type is "concept completion"
The question concept is:
((ACTOR HUM1 <=> (*MTRANS*) MOBJECT
    (*CONCEPTS* CONCERNING CNTRCT2) TO (*?*)) TIME G1062)
   applying INFER-FROM-TOPIC to CNTRCT2
      inferring undifferentiated political meeting
The inferred question concept is:
((<=> (sM-MEETING ACTOR HUM1 OTHERS (*?*) TOPIC CNTRCT2))
   TIME G1062)
searching memory for question concept
  searching directly for sM-MEETING
   found (GN545 GN525 GN517 GN486 GN453)
  applying FIND-FROM-STANDARDIZATIONS to search for
          standard types of sM-MEETINGs
     searching for $MEET
     searching for $CONSULT
     searching for $PUB-REL-MEET
      found (GN590 GN556 GN561)

Carter, Brezhnev, Gromyko, and other American and
Russian diplomats
```

In searching for situations where Vance talked about SALT, CYRUS first infers that those discussions would have had to have been at political meetings. It then searches memory for instances of the kinds of political meetings it knows about -- "diplomatic meetings" about SALT ($MEET), consultations about SALT with American government officials ($CONSULT), and public relations meetings about SALT ($PUB-REL-MEET). It finds meetings of each type with a number of people -- consultations

with Carter and other American diplomats, and diplomatic meetings with Gromyko, Brezhnev, and other Russian diplomats.

Another E-MOP with standard methods of implementation that CYRUS knows about is "ceremonies" (sM-CEREMONY). In trying to recall ceremonies, then, one strategy CYRUS applies is Find-from-standardizations. Using this strategy, CYRUS searches for welcomes ($WELCOME), parades ($PARADE), and state dinners ($STATE-DINNER) individually as follows.

```
Enter next question:

What ceremonies have you taken part in?

The question is:
((<=> (sM-CEREMONY ACTOR HUM1 SPEC (*?*))) TIME G1057)
The question type is "concept completion"
The question concept is:
((<=> (sM-CEREMONY ACTOR HUM1 SPEC (*?*))) TIME G1057)
searching memory for question concept
  searching directly for sM-CEREMONY
   found (GN300)
  applying FIND-FROM-STANDARDIZATIONS to search for
         standard types of sM-CEREMONYs
     searching for $WELCOME
     searching for $PARADE
     searching for $STATE-DINNER
      found (GN363 GN300 GN600 GN452 GN586 GN345)

A state dinner in Israel, a parade and state dinner
in England, and a number of welcoming ceremonies.
```

Notice that by using this strategy, CYRUS can only retrieve standard types of ceremonies. In fact, one of the ceremonies Vance has taken part in, which was not retrieved using this strategy, was a ceremony commemorating the opening of commercial air traffic between Israel and Egypt. The ceremony consisted of taking off from Beer Sheba in Israel, flying over Cairo and Tel Aviv airports, and landing back in Beer Sheba. Because this was not a standard type of ceremony, CYRUS could not retrieve it using this strategy. In order to retrieve that ceremony, CYRUS would have to use other strategies to infer reasons why ceremonies might have taken place recently and what the form of those ceremonies might be.

As shown by CYRUS' failure above, one of the consequences of using event-oriented search strategies is that they can direct search only for standard types of related contexts. Because these strategies make use of content frame information, which is generalized or standard information about different types of episodes, unusual related events cannot be retrieved using these strategies. They are useful, however,

for finding more common types of episodes. More unusual events can be found in other ways -- by using *enumeration strategies*, *for example*, to imagine possibilities, or by searching for interruptions or other standard ways that events can fail.

4.3.3 Using larger episodes

Most E-MOPs hold information about the types of larger episodes they are normally part of. For example, meetings generally occur in the context of negotiations, and sightseeing usually happens during trips. A person answering a question about sightseeing in a particular place is quite likely to try to remember a trip to that place. To remember meetings, CYRUS attempts retrieval of relevant negotiations episodes the meetings could have been part of.

There are two strategies that can guide this processing -- Find-from-IMOP and Find-from-SMOP. Find-from-IMOP is used to search for IMOP-related episodes an event could have been part of. An IMOP (Schank, 1980) is a goal or intention-based type of episode which takes place over a long period of time. A vacation, for example, is an IMOP. It is done for relaxation. Negotiating is also an IMOP. It is done for the purpose of solving an important problem. Because an IMOP episode takes place over a long period of time, its sequence of events is not as important as the few particular events in its sequence of events which *stand out as* contributing to obtaining its goal. Thus, a "negotiations" episode does not refer to all of the meetings in its sequence of events, but only to those that were significant.

Find-from-SMOP is used to search for larger episodes in whose sequence of events a target event might have been. An SMOP (Schank, 1980) or simple MOP is a situation whose sequence of events is fairly stereotyped, and which can include other simple MOPs and scripts (Schank and Abelson, 1977) in its sequence of events. In both of these strategies, appropriate larger episodes the target concept could have been part of are inferred (using Instantiate-larger-episodes), and their contexts are searched to see if they could have included the target concept.

In searching memory for meetings with Begin, for example, CYRUS attempts retrieval of negotiations episodes involving Israel. After retrieving one of those episodes, CYRUS searches its sequence of events to see if it included a particular meeting with Begin. If it did not specify such a meeting, CYRUS can use information in the negotiations episode to hypothesize a meeting with Begin that would have occurred in its context, in this way, perhaps, better specifying the target concept. Memory could then be traversed searching for the newly-expanded target event. The meeting hypothesized from the negotiations episode would match the target concept but contain more information, since it was constructed from a particular negotiations context. Any meeting found in memory that matched it would also match the target concept. These two strategies work as follows:

Find-From-IMOPs

IF the Target Concept could have been dominated by an Intentional sequence (IMOP)

THEN

(1) Use "Instantiate-Larger-Episodes" to hypothesize IMOP episodes the target concept could have been part of. Use Component-Instantiation strategies to fill in each description.

(2) Search memory for those instances.

(3) If one is found, search its goal progressions for instrumental events which could match the Target Concept.

(4) If no match is found, use "Instantiate-Seq-of-Events", plus the additional information found in the IMOP instance, to better specify the target concept. Search for the new target concept in memory.

Figure 4-4

Find-From-simple-MOPs

IF the Target Concept could have been dominated by a larger episode (sMOP)

THEN

(1) Use "Instantiate-Larger-Episodes" to hypothesize simple MOPs the target concept could have been part of. Use Component- Instantiation strategies to fill in each simple MOP description.

(2) Search memory for that instance.

(3) If one is found, search its known sequence of events for a match to the target concept.

(4) If no match is found, use "Instantiate-Seq-of-Events", plus the additional information found in the IMOP instance, to better specify the target concept. Search for the new Target Concept in memory.

Figure 4-5

Notice that there are two search strategies guiding search for larger episodes an event could be part of while there is only one instantiation strategy (Instantiate-Larger-Episodes) necessary to actually construct these larger episodes. Larger episodes, whether simple MOP episodes or IMOP episodes, have the same containment relationship to smaller episodes they contain. Thus, only one instantiation strategy is necessary.

On the other hand, the relationship of an event to simple MOPs is merely a containment relation through the sequence of events, while the relationship of an event to IMOPs it is part of is instrumental. The smaller episode is instrumental to obtaining the goals of the larger intention-related episode. Meetings are instrumental to obtaining the goals of negotiations. They are merely part of the sequence of events of a diplomatic trip. Because IMOPs are closely related to goals, application of Find-from-IMOP is a way of implicitly using possibie goals of an episode to search memory.

In terms of processing, Find-from-simple-MOPs and Find-from-IMOPs are different in steps 3 and 4. Those are the steps that guide search of the episodic context of an event. In Find-from-simple-MOPs, the sequence of events of the simple MOP instance found can be checked or reconstructed to determine if it contained an event matching the target

concept. Because IMOPs take place over long periods of time, and because their sequences of events are not as important as the changes in their goals, their sequences of events will not be well-filled-out. Events instrumental to obtaining their goals will have to be found by remembering or reconstructing their goal progressions and remembering the events that prompted those changes.

CYRUS applies the strategy Find-from-simple-MOPs whenever the type of episode it is looking for specifies which kinds of simple MOP episodes it is usually part of. "Diplomatic meetings" are often included in "conferences". "Sightseeing" is usually part of a trip. Thus, in searching extensively for sightseeing experiences, CYRUS searches for trips to the appropriate places.

```
Enter next question:

>Have you ever been sightseeing in Egypt?

The question is:
((<=> (sM-SIGHTSEE ACTOR HUM1)) TIME G0874 MODE (*?*)
   PLACE POL6)
The question type is "verification"
The question concept is:
((<=> (sM-SIGHTSEE ACTOR HUM1)) TIME G0874 PLACE POL6)
searching memory for question concept
  searching directly for sM-SIGHTSEE
       didn't find any
applying strategies to search memory
    ...
  applying FIND-FROM-SIMPLE-MOPS to search for
    episodes sM-SIGHTSEE could have occurred in
     searching for sM-VIPVISIT
        found (GN528 GN536 GN551 GN407)
      searching sM-VIPVISIT instance for input
      searching sM-VIPVISIT instance for input
      searching sM-VIPVISIT instance for input
      searching sM-VIPVISIT instance for input
        didn't find any sM-SIGHTSEE
    ...

No.
```

After failing to recall sightseeing experiences in Egypt, CYRUS applies search strategies to attempt to retrieve sightseeing experiences in some other way. Why would searching for trips help CYRUS to find sightseeing experiences if it could not find them individually? In searching for "sightseeing in Egypt", CYRUS could not find an episode, perhaps because they were not indexed by country and that was all the information it had. If a well-specified trip to Egypt or the Middle East could be found, however, it might refer to sightseeing done during

the trip. CYRUS attempts to retrieve trips to Egypt by constructing a trip context and traversing appropriate E-MOPs. As it retrieves each one, it searches its sequence of events for a sightseeing episode. It doesn´t find any.

In searching extensively for diplomatic meetings, CYRUS applies Find-from-simple-MOPs to search for appropriate conferences the meeting could have been part of. Output from CYRUS follows:

```
Enter next question:

>What have you talked about with Gromyko?

The question is:
((ACTOR HUM1 <=> (*MTRANS*) MOBJECT
     (*CONCEPTS* CONCERNING (*?*)) TO HUM66) TIME G1548)
The question type is "concept completion"
The question concept is:
((ACTOR HUM1 <=> (*MTRANS*) MOBJECT
     (*CONCEPTS* CONCERNING (*?*)) TO HUM66) TIME G1548)
   applying INFER-FROM-PARTICIPANTS to HUM66
     inferring a diplomatic meeting
The inferred question concept is:
((<=> ($MEET ACTOR HUM1 OTHERS HUM66 TOPIC (*?*)))
   TIME G1548)
searching memory for question concept
  searching directly for $MEET
   found (GN453)
applying strategies to search memory
    ...
  applying FIND-FROM-SIMPLE-MOPS to search for
   episodes $MEET could have occurred in
     searching for sM-SUMMIT-CONFERENCE
     searching for sM-CONFERENCE
     searching for sM-VIPVISIT
        found (GN540 GN481)
      searching sM-VIPVISIT instance for input
      searching sM-VIPVISIT instance for input
        found (GN561 GN564 GN567 GN485 GN488 GN489)
    ...

SALT and other arms limitations topics.
```

After first inferring that Vance would have talked to Gromyko at diplomatic meetings, CYRUS searches directly for those meetings. When memory returns the message that there are numerous meetings but more specification is needed to retrieve them, CYRUS attempts to retrieve them by applying search strategies. Thus, it applies Find-from-simple-MOPs to search for conferences and diplomatic trips meetings with Gromyko could be part of. It calls

Instantiate-larger-episodes to instantiate "summit conferences about Russian concerns with Gromyko" and "diplomatic trips to Russia", and searches memory for each of those. After find those contexts, it retrieves the meetings with Gromyko that they included.

The strategy Find-from-IMOPs is similar. In applying Find-from-IMOPs, CYRUS applies Instantiate-Larger-Episode to construct an appropriate IMOP episode, traverses memory searching for that episode, and then searches the context of those episodes for the target. Because memory does not hold many IMOP episodes (because they are longer range), and because IMOP episodes do not have as well filled-out sequences of events as simple MOPs, CYRUS generally finds more of the existing relevant IMOP episodes than it would simple MOP episodes, but does not always find instances of the target concept within those episodes.

One of the IMOPs CYRUS knows about is "negotiations" (I-NEGOTIATE). Diplomatic trips are usually in the service of negotiations. Thus, one strategy CYRUS applies to search for diplomatic trips to Israel is Find-from-IMOPs.

```
Enter next question:

>How many times have you been to Israel?

The question is:
((ACTOR HUM1 IS (*LOC* VAL POL5)) TIME G0658 QUANTITY (*?*))
The question type is "concept completion"
The question concept is:
((ACTOR HUM1 IS (*LOC* VAL POL5)) TIME G0658)
    applying INFER-FROM-LOCATION to POL5
       inferring diplomatic trip context
The inferred question concept is:
((<=> (sM-VIPVISIT ACTOR HUM1 DESTINATION POL5)) TIME G0658)
searching memory for question concept
  searching directly for sM-VIPVISIT
   found (CON131)
applying strategies to search memory
  applying FIND-FROM-IMOPS to search for episodes
          $MEET could have occurred in
     searching for I-NEGOTIATE
        found (GN524 GN522)
      searching I-NEGOTIATE instance for input
        found (GN533 GN546)
  ...

Twice to negotiate the Camp David Accords, and a few other
times to negotiate Middle East peace.
```

After first inferring that Vance would have been to Israel on a diplomatic trip, CYRUS searches memory for those trips. Because there are many, it recalls only the most recent and then applies search strategies to find others. Because diplomatic trips are normally in the service of negotiations, it applies Find-from-IMOPs to search for negotiations episodes they could have been part of. CYRUS retrieves negotiations about the Camp David Accords and other negotiations about Middle East peace, and recalls from their contexts a number of diplomatic trips.

4.3.4 Using preceding events

Enablements, preconditions, causations, and temporally preceding events of a target event all precede the target event. Enablements are the episodes and states that allow a state to be in existence or an event to happen. Enablements for Vance's being secretary of state include his being appointed by the President and approved by the Senate. Enablements for being a lawyer include going to law school and passing the bar exam.

Preconditions are states that must be true in order for an event to occur. The preconditions for a meeting are that all participants are present (unless it is a conference call). Preconditions for a successful telephone call are that participants be in different places, that they not be on the phone with someone else and that the person being called is in close proximity to the telephone being called, and that the phone lines be working.

Causations of an event are events which caused the target event to happen or states that require that an episode be done. When Vance is told to go to Russia to negotiate SALT with Gromyko, the fact that he is told to go by Carter is one of the reasons he goes on the trip, thus one of the trip's causations.

Events can be searched for in memory by attempting to recall preconditions, enablements, causations, or preceding events. To answer "Have you ever been to a summit conference in Belgium?", Cyrus Vance might attempt to recall reasons why such a summit conference would have been called. Reasons for summit conferences include the existence of international world problems that have to be solved. Recalling those problems might aid recovery of summit conferences.

The strategy "Find-from-Reasons" attempts to retrieve a target concept by hypothesizing the causations or reasons for doing that event and searching for instances of those reasons. If they can be found, their resulting events might match the target concept. Alternatively, reasons can be used to further fill out the target concept, and the new fuller concept can be searched for. If Vance were trying to retrieve summit conferences he had attended in Belgium, he might try to retrieve reasons the summit conferences would have been called. Major world problems are reasons for summit conferences, so he might try to remember major world problems and see if any of them led to summit conferences. If he couldn't find any summit conferences that way, he might use

information about world problems to infer possible topics for summit conferences and try again to retrieve them -- this time with a better description of what must be retrieved.

Find-From-Reasons

IF the Target Concept's E-MOP specifies reasons why the Target Concept could have been done,

THEN

(1) Apply "Instantiate-Reasons" to infer reasons why the Target Concept could have happened.

(2) Search for them.

(3) If any are found, see if any of their resulting events or the events following them match the Target Concept.

(4) If reasons are found but their resulting events are not found, then use the information on the reason to instantiate a result that matches the Target Concept. Search for it in memory. If a match is found, it will also match the target.

Figure 4-6

After finding the kinds of events which could cause the target event to happen, the event contexts of the events found are searched for a match to the target concept (step 3). Aspects of the episodic contexts of events found that are searched are the events it caused and events which followed it.

Often, retrieving activities that might have preceded an event can aid in its retrieval. Since welcoming ceremonies normally come after flights during diplomatic trips, a particular welcoming ceremony might be retrieved by finding the flight that preceded it. For example, to retrieve "when the welcoming ceremony for Vance at the airport in Spain started three hours later than planned", it would probably be appropriate for a memory with events from Vance's point of view to attempt retrieval of a flight to Spain that was delayed for three hours. The strategy "Find-from-Preceding-Events" is used to retrieve an event based on events that preceded it. Enablement conditions, preconditions, and typical preceding events can be used similarly to causations to find a target concept. "Find-from-Preceding-Events" is similar to "Find-from- Reasons" except that "Instantiate-Preceding-Events" is called in step (1). Similarly, "Find-from-Preconditions" calls "Instantiate-Preconditions" in step (1). "Find-from- Enablements" calls

"Instantiate-Enablements" in step (1). Both "Find-from-Enablements" and "Find-from-Preconditions" call "Instantiate-Enabled-Events" and "Instantiate-Results" in step (4). Those strategies are presented below.

Find-From-Preceding-Events

IF the Target Concept's E-MOP specifies typical preceding events

THEN

(1) Instantiate those preceding events using "Instantiate-Preceding-Events", and fill in the components using Component-Instantiation strategies.

(2) Search memory for those preceding events.

(3) If any are found, see if any of their resulting events or the events following them match the target concept.

(4) If preceding events are found but their resulting or following events are not, then use the information on what is found to instantiate a result or succeeding event that matches the target concept. Search for it in memory. If a match is found, it will also match the target.

Figure 4-7

Find-From-Preconditions

IF the Target Concept's E-MOP specifies preconditions

THEN

(1) Instantiate those preconditions using "Instantiate-Preconditions", and fill in the components using Component-Instantiation strategies.

(2) Search memory for the preconditions.

(3) If any are found, see if any of their resulting events or the events following them match the Target Concept.

(4) If preconditions are found but their resulting or following events *are not*, then use the information on what is found to instantiate a result that matches the Target Concept. Search for it in memory. If a match is found, it will also match the target.

Figure 4-8

Find-From-Enablements

IF the Target Concept's E-MOP specifies enablement conditions

THEN

(1) Instantiate those enablement conditions using "Instantiate-Enablements", and fill in the components using Component-Instantiation strategies.

(2) Search memory for the enablement conditions.

(3) If any are found, see if any of their resulting events or the events following them match the Target Concept.

(4) If enablements are found but their resulting events are not, then use the information on what is found to instantiate a result that matches the Target Concept. Search for it in memory. If a match is found, it will also match the target.

Figure 4-9

4.3.5 Finding a target from resulting events

The final set of search strategies are those that guide search for events resulting from or following the target event. Resulting events include events enabled by the target event, and events and states caused by the target event. A student or ex-student might recall tests he's taken by remembering extraordinary results -- the time he aced an exam or the one time he flunked, the time the results of an exam got him into a job he wanted, etc. It would be appropriate for CYRUS to recall individual negotiations episodes by remembering possible results of those episodes and the negotiations they belonged to. International negotiations can end successfully with the signing of a treaty or pact, can end unresolved but low key, or can end unresolved and cause war. One way to find a meeting between Vance and Carter during which Carter told Vance to go to the Middle East would be to search for trips by Vance to the Middle East, the event that would result from the meeting.

Thus, a target concept can sometimes be found by searching for its probable results and checking any results found to see if their preceding events or reasons match the target concept. The strategy that

would direct that search is "Find-From-Results", described below.

Find-From-Results

IF the Target Concept's E-MOP specifies normal results

THEN

(1) Instantiate those results using "Instantiate-Results", and fill in the components using Component-Instantiation strategies.

(2) Search memory for the results.

(3) If any are found, see if any of the events preceding them or their reasons match the Target Concept.

(4) If resulting events are found but their reasons or preceding events are not, then use the information on what is found to instantiate an event that matches the Target Concept. Search for it in memory. If a match is found, it will also match the target.

Figure 4-10

Searching for events and states that could be enabled by the target event or that could have followed the target event can also aid retrieval. A good way to answer the question "Did Vance go to law school?", if it were not known, would be to recall whether he had ever been a lawyer. If so, then in the normal case, he would have gone to law school. The strategy to guide this search is "Find-from-Enabled-Events". It guides instantiation and retrieval of events the target event could have enabled, then uses those events to find a match in memory to the the target event by checking to see if the events which caused, enabled, or preceded those events match the target. Thus, step 3 of this strategy is similar to step 3 of Find-from-Results.

Find-From-Enabled-Events

IF the Target Concept's E-MOP specifies events or states typically enabled by events in that E-MOP

THEN

(1) Instantiate those events or states using "Instantiate-Enabled-Events", and fill in the components using Component-Instantiation strategies.

(2) Search memory for the enabled events.

(3) If any are found, see if any of the events enabling them or events preceding them match the Target Concept.

(4) If enabled events are found but their enablements or preceding events are not, then use the information on what is found to instantiate an event that matches the Target Concept. Search for it in memory. If a match is found, it will also match the target.

Figure 4-11

Events can follow other events in time but not be related causally. This often happens in scriptal situations (Schank and Abelson, 1977). In a restaurant, being seated usually precedes ordering, but does not actually enable ordering in the strict sense of the word. When events that could follow a target concept can be inferred, the strategy Find-from-Following-Events can be used to search memory for the target. Like Find-from-Enabled-Events and Find-from-Results, reasons, enabling events, and preceding events of the related event's episodic context are checked or reconstructed to see if they could match the target event.

Find-From-Following-Events

IF the Target Concept's E-MOP specifies events that usually follow it in time

THEN

(1) Instantiate those events using "Instantiate-Results", and fill in the components using Component-Instantiation strategies.

(2) Search memory for those following events.

(3) If any are found, see if any of the events preceding them, enabling them, or causing them match the Target Concept.

(4) If following events are found but their reasons, enablements, or preceding events are not, then use the information on what is found to instantiate an event that matches the Target Concept. Search for it in memory. If a match is found, it will also match the target.

Figure 4-12

4.4 Implications

One important implication of search strategy application is that retrieval is fallible. There is no guarantee that all episodes being looked for in memory will be retrieved. If, during processing, a strategy which would yield an answer is neglected, episodes which would have been found by that strategy might not be found.

In a previous example, CYRUS was shown to be fallible in this regard in its inability to recall a ceremony commemorating the opening of air traffic routes by flying between the two airports. If flying had been a standard sort of ceremony, that episode could have been recalled by recalling flying ceremonies. Flying, however, is not standard, so the ceremony could not be recalled in that way.

During extensive memory search, strategies can be applied in succession until a sufficient answer is found. Perhaps another strategy could be used to recall this ceremony. If the reason for the ceremony could be retrieved, it could be used to remember this ceremony. Again, however, opening air routes is not a standard reason for a diplomatic

ceremony and may be hard to retrieve. On the other hand, if in answering this question, ceremonies involved with negotiations could have been recalled, perhaps this particular ceremony could have been retrieved. Failure to apply a relevant strategy, then, can result in failure to retrieve relevant information from memory.

In extensive memory search, these strategies can be applied in succession until a satisfactory answer is retrieved from memory. When people attempted to recall museum experiences (see chapter 2), they seemed to apply search strategies in succession. In his observations of people recalling persons in their high school class, Williams (1978) reported application of a succession of search strategies until a sufficient answer was found. In attempting to recall meetings with Gromyko, CYRUS recalls trips to Russia, negotiations episodes negotiating SALT, and summit conferences about SALT and arms limitations. Memory search can continue until strategies stop producing results or until a sufficient answer is found.

When CYRUS searches its memory extensively, it applies each appropriate strategy in succession until it has found a satisfactory answer. CYRUS knows it has found a satisfactory answer when it is asked for only one episode, and it finds one (as when trying to verify if something has happened); when it is asked for a specific number of episodes and it finds that many episodes; or when it is finished applying all the strategies it knows about. Thus, when searching for all of Vance's meetings with Gromyko, CYRUS first uses context-instantiation strategies to infer that Vance and Gromyko must have talked at diplomatic meetings. Thus, it confines its search to diplomatic meetings. It then searches for diplomatic meetings with Gromyko, and applies strategies to search for simple MOP instances and IMOP instances during which meetings with Gromyko could have occurred. Output from CYRUS follows:

```
Enter next question:

>What have you talked about with Gromyko?

The question is:
((ACTOR HUM1 <=> (*MTRANS*) MOBJECT
    (*CONCEPTS* CONCERNING (*?*)) TO HUM66) TIME G1548)
The question type is "concept completion"
The question concept is:
((ACTOR HUM1 <=> (*MTRANS*) MOBJECT
    (*CONCEPTS* CONCERNING (*?*)) TO HUM66) TIME G1548)
   applying INFER-FROM-PARTICIPANTS to HUM66
     inferring a diplomatic meeting
The inferred question concept is:
((<=> ($MEET ACTOR HUM1 OTHERS HUM66 TOPIC (*?*)))
   TIME G1548)
searching memory for question concept
  searching directly for input -- $MEET
   found (GN453)
applying strategies to search memory
  applying FIND-FROM-SIMPLE-MOPS to search for
   episodes $MEET could have occurred in
     searching for sM-SUMMIT-CONFERENCE
     searching for sM-CONFERENCE
     searching for sM-VIPVISIT
        found (GN540 GN481)
      searching sM-VIPVISIT instance for input
        found (GN561 GN564 GN567)
      searching sM-VIPVISIT instance for input
        found (GN485 GN488 GN489)
  applying FIND-FROM-IMOPS to search for episodes
          $MEET could have occurred in
     searching for I-NEGOTIATE
        found (GN391A)
      searching I-NEGOTIATE instance for input
        found (GN542 GN594)

SALT and other arms limitations topics.
```

In searching memory to find out who Vance has discussed SALT with, CYRUS first decides those discussions were at political meetings and then applies Find-from- Standardizations, Find-from-simple-MOPs, and Find-from-IMOPs to find them -- the strategies it finds appropriate information for in its "political meetings" MOP.

```
Enter next question:

>Who have you talked to about SALT?

The question is:
((ACTOR HUM1 <=> (*MTRANS*) MOBJECT
    (*CONCEPTS* CONCERNING CNTRCT2) TO (*?*)) TIME G1062)
The question type is "concept completion"
The question concept is:
((ACTOR HUM1 <=> (*MTRANS*) MOBJECT
    (*CONCEPTS* CONCERNING CNTRCT2) TO (*?*)) TIME G1062)
   applying INFER-FROM-TOPIC to CNTRCT2
      inferring undifferentiated political meeting
The inferred question concept is:
((<=> (sM-MEETING ACTOR HUM1 OTHERS (*?*) TOPIC CNTRCT2))
   TIME G1062)
searching memory for question concept
  searching directly for input -- sM-MEETING
   found (GN545 GN525 GN517 GN486 GN453)
applying strategies to search memory
  checking locative preconditions of input
   could have happened during business trip to USSR
    searching for business trips to POL15
     found (GN481 GN390)
      searching sM-VIPVISIT instance for input
        found (GN485 GN488 GN486 GN489)
      searching sM-VIPVISIT instance for input
  applying FIND-FROM-IMOPS to search for episodes
          sM-MEETING could have occurred in
     searching for I-NEGOTIATE
        found (GN391A GN486)
      searching I-NEGOTIATE instance for input
        found (GN525 GN545 GN488)
  applying FIND-FROM-SIMPLE-MOPS to search for
   episodes sM-MEETING could have occurred in
     searching for sM-CONFERENCE
        found (GN448)
      searching sM-CONFERENCE instance for input
        found (GN450 GN453 GN456)
  applying FIND-FROM-STANDARDIZATIONS to search for
          standard types of sM-MEETINGs
     searching for $MEET
     searching for $CONSULT
     searching for $PUB-REL-MEET
      found (GN590 GN556 GN561)

Carter, Brezhnev, Gromyko, other American and Russian
diplomats, and Egyptian vice president Mustafa Khalil.
```

Because CYRUS′ memory organization provides multiple references to the same event, CYRUS often finds the same answer by following multiple paths. In the example above, some of the same meetings were retrieved by searching for trips that were retrieved by searching for negotiations.

4.5 Summary

Search strategies are used in searching memory extensively to find a target event. They guide search for event contexts related to the target -- events that may have happened if the target event had happened, and which may refer to an instance of the target. After instantiating and retrieving an event that could contain the target event in its context, they search appropriate parts of its episodic context to find an instance of, or reference to, the target event.

Search strategies have 4 steps. They call instantiation strategies to construct an alternate context for search from a target concept. They guide the traversal process by giving it a context specification to look for. If the alternate context for search is found, search strategies guide search of its episodic context to find the target concept. If that does not produce a match to the target, they guide reconstruction of the appropriate aspect of the event′s episodic context. Thus, they guide both application of instantiation strategies and search of an events′ episodic context during extensive memory search.

Failure to apply a strategy will often result in failure to retrieve a relevant event from memory. In later chapters, we will explore how search strategies can be chosen during extensive memory search, and under what other circumstances retrieval strategies are used.

CHAPTER 5

Organization of Generalized Information in Memory

5.1 Introduction

Previous chapters described retrieval strategies useful for searching conceptual memory. We have been assuming that the generalized knowledge those strategies needed was resident in memory, but have not yet described its organization. In this chapter, the kinds of generalized information retrieval strategies require and how that information is organized will be described.

5.2 CYRUS' E-MOPs

CYRUS' E-MOPs provide retrieval strategies with the knowledge they need to operate by describing their connections to other E-MOPs, their usual components, and the relationships of their components to components of related E-MOPs. The following chart lists CYRUS' E-MOPs.

CYRUS' E-MOPs

$MEET -- diplomatic meetings with foreign dignitaries
$CONSULT -- consultations with American officials
$PUB-REL-MEET -- political meetings with non-diplomats
$STATE-DINNER -- state dinners
$FLY -- flights
$REPORT -- press conferences
$SPEECH -- speeches

sM-MEETING -- political meetings
sM-CONFERENCE -- political conferences
sM-SUMMIT-CONFERENCE -- specialized political conferences
sM-TRAVEL -- traveling
sM-TRIP -- generalized trip
sM-VIPVISIT -- diplomatic trip
sM-SIGHTSEE -- sightseeing
sM-SOCIAL-POL-ACTIVITY -- social political activities
sM-DIPLOMATIC-ACTIVITY -- diplomatic activities

I-NEGOTIATE -- negotiations
I-DIPLOMACY -- diplomacy
I-SOCIALIZE -- socializing

Figure 5-1

CYRUS' E-MOPs are related to each other through the following causal, temporal, containment and specialization relationships:

Relationships between E-MOPs

Causal relationships
- --usual events and states that enable these events to happen
- --their normal preconditions
- --events and states that usually caused these events to happen
- --events and states these events usually cause
- --events and states these events usually enable

Containment relationships
- --E-MOPs these events are usually part of
- --the typical sequence of events for these episodes

Temporal relationships
- --what kinds of events usually come before these episodes
- --what kinds of events usually follow these episodes
- --what kinds of events usually go on at the same time

Specialization/generalization relationships
- --parent E-MOPs
- --standard methods of achievement

Figure 5-2

The figure below shows some of the interconnections between CYRUS′ E-MOPs.

The relationships between CYRUS′ E-MOPs

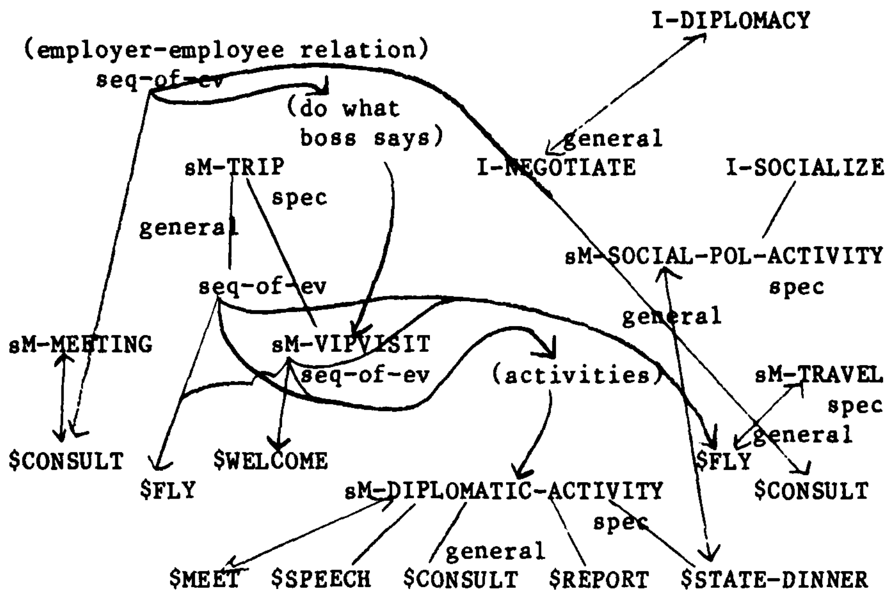

Figure 5-3

The sequence of events for a trip (sM-TRIP) includes flying, doing trip activities and flying home. Its specialized E-MOP "diplomatic trip" (sM-VIPVISIT) includes, in addition to that, a welcoming ceremony. Instead of general trip activities, it specifies diplomatic activities. The reasons for Vance′s and Muskie′s diplomatic trips usually stem from some world problem (not shown) which initiates a consultation with Carter in which Carter advises travel. When the trip is done, the Secretary of State reports back. We can think of that as a generalized sequence of events describing an "employer-employee relationship" (at the top).

In addition to their relationships to other E-MOPs, E-MOPs must specify default values for their components. The following are the kinds of components CYRUS′ E-MOPs can have:

Role-fillers for E-MOPs

1. animate participants -- people, countries, organizations, etc.
2. inanimate participants -- associated objects
3. topic of discussion
4. location
5. time
6. duration

Figure 5-4

Each of CYRUS' E-MOPS specifies both the other MOPs it is related to and the relationships between its components. CYRUS' "diplomatic meetings" MOP, for example, specifies each of the E-MOPs it is related to, what its components look like, and how each of its components correspond to components of other E-MOPs it is related to. "Diplomatic meetings" ($MEET) are specialized "political meetings" (sM-MEETING), as are its "consultations" and "public relations meetings". A "political meeting" has a sequence of events and typical setting common to all meetings. In addition to those attributes which it inherits from "political meeting", "diplomatic meeting" specifies that its participants be foreign diplomats, that it is usually part of "negotiations", that they happen during "conferences" and "diplomatic trips", that their topic of discussion is usually an international contract, etc.

Below is the generalized information CYRUS has about "diplomatic meetings" ($MEET).

```
  "diplomatic meetings" ($MEET) -- generalized information

larger episodes:
  "negotiations" (I-NEGOTIATE)
      participants = diplomats of same nationality as
                     meeting participants, and of other
                     countries involved in meeting topic
      topic = generalization of meeting topic
  "diplomatic trips" (sM-VIPVISIT)
      destination = country or area involved in meeting
                    topic, or country of residence of
                    meeting participant
   "conferences" (sM-CONFERENCE)
       participants = diplomats of same nationality as
                      meeting participants, and of other
                      countries involved in meeting topic
       location = location of meeting
       topic = generalization of meeting topic

enables:
   "treaty signing" ($TREATY)
       sides = sides of meeting topic

sequence of events:
   participants MTRANSing to each other about meeting topic

preceding and following events:
   "diplomatic meetings" ($MEET)
       participants = subset of meeting participants
       topic = aspect of meeting topic

more general E-MOPs and classifications:
   "political meetings" (sM-MEETING)
        all components correspond
   "diplomatic activities"
        all components correspond

role fillers:
    participants:  foreign dignitaries of countries involved
                   in contract being discussed
    location:  conference room in capital city of country of
               residence of some important participant
    topic:  international contract
    duration:  one to two hours
```

Figure 5-5

Like "diplomatic meetings", "public relations meetings" ($PUB-REL-MEET) is also a specialization of "political meeting", but has different specialized features than does diplomatic meeting. Their

participants are representatives of lobby groups and are non-politicians, while for "diplomatic meetings", the participants are foreign diplomats. In addition, public relations meetings happen during important negotiations, but do not usually have direct bearing on those negotiations.

Notice that the relationships between E-MOPs cannot be stored as simple links. Rather, they must also specify correspondences between their components. Thus, CYRUS' "diplomatic meetings" MOP specifies not only that it is related to "negotiations", but that the participants in the "negotiations" it is related to are the countries the participants of the meetings represent, and that the topic of the negotiations includes the topic of the meeting.

5.3 Information necessary for context construction

All information used by retrieval strategies does not come from E-MOPs. In particular, the context instantiation strategies which construct contexts for search in the initial step of the retrieval process do not use E-MOP information at all -- in fact, they do not have E-MOP information available to them.

Recall the following two questions, discussed previously:

(Q5-1) Who have you talked to about SALT?

(Q5-2) Has your wife ever met Mrs. Begin?

In neither of these is an E-MOP for traversal mentioned. Using specifications that are given in the question, context instantiation strategies must infer an E-MOP for traversal. In the first question, only the international contract "SALT" is available to aid context construction. In the second question, the two people mentioned, their relationship, and Vance's relationship to them can be used for context construction.

To answer (Q5-1), the description of SALT must be used to infer a search context, or E-MOP for traversal. "SALT", as a concept in memory, has no contexts associated with it. "SALT", however, is defined in memory as an "international contract", which does have contexts associated with it. An "international contract" can be talked about in a "political meeting" (sM-MEETING). Using that information, a "political meeting" context with topic "SALT" can be constructed and searched for.

There are two important points about the organization of concepts in memory that are shown in this example. First, context construction requires <u>association of an E-MOP specification with a non-event concept</u> in memory. In order to infer a context for "talking about SALT", "international contract" had to have an E-MOP associated with it. If it had no references to E-MOPs, context construction could not have happened.

Second, a *hierarchy of concepts* will allow contexts to be associated with more general concepts than those specified in a question, and through *inheritance* of properties, an E-MOP can be inferred from those more general concepts. Although SALT had no contexts associated with it, the contexts associated with "international contract" could be used.

An important conclusion to draw from these two observations is that context construction can proceed in exactly the same way whether a particular entity or a class it falls into is specified in a question. Compare, for example, the following two questions:

(Q5-3) Who have you talked to about the Camp David Accords?

(Q5-4) Who have you talked to about international contracts?

The "Camp David Accords", like "SALT", is an "international contract". Thus, using the information that "international contracts" are talked about at "political meetings", the target concept "political meetings about the Camp David Accords" can be inferred. In answering (Q5-4), information from the concept "international contract" is also used, and the target concept "political meetings about international contracts" can be inferred. The only difference here is that after infering a context in (Q5-3), the specific contract mentioned in the question is filled in.

Thus, in exactly the same way "political meeting about SALT" can be inferred from the original target "talking about SALT" in (Q5-2), a context can be constructed for answering (Q5-3) and (Q5-4). The following hierarchical organization of contracts in memory allows that construction:

```
                    "international contracts"

               "international contracts"
                   E-MOPs:  "political meetings" topic
                   sides:  countries
               /\                     /\
              /                         \
             / general                   \  general
            /                             \
     "SALT"                        "the Camp David Accords"
       sides: Russia, the US         sides: Israel, Egypt
       topic: arms limitations       topic: peace
```

Figure 5-6

"International contracts" have countries as their sides and are the topics of political meetings. "SALT" is an "international contract" with sides Russia and the United States, and underlying issue "arms

limitations". "The Camp David Accords" is another "international contract" with underlying issue peace and involving Israel and Egypt. Other international contracts in memory can be represented similarly.

In addition to this organization of particular contracts as international contracts, each of the particular contracts above might belong to other classifications of contracts. "The Camp David Accords", for example, is a "peace contract" (a contract with underlying issue peace), and if there were any contexts associated with "peace contract", those also could be used during context construction. It is also a specialization of "Arab Israeli peace", also an "international contract", and if there were contexts associated with that classification of contracts, those too would be used during context construction.

The actual hierarchies of concepts there are in memory depends on how the concepts are used. Thus, unless "peace contract" has some necessary generalized information for processing associated with it, it does not need to be an explicit classification in the contract organization, although there may be many examples of peace contracts among the particular contracts organized there.

Similarly, the types of concepts which have E-MOPs associated with them should be those concepts (and classes of those concepts) which are associated with events. In CYRUS' memory, that includes animate participants, such as people and organizations, inanimate participants, such as objects, locations, and times. In addition, the classifications each particular person, country, or object fits into should have E-MOPs associated with them.

In a political context, people can be described by the political roles they play, their nationalities, their affiliations to political groups, and their occupations. Thus, generalized information about people and classifications of people should be based around those specifications. Menachim Begin, for example, is "Israeli", a "prime minister", and acts as a "delegate" or "negotiator". A "prime minister" is a "head of state". "Heads of state", and most other high political officials are "diplomats". "Negotiators", too, are "diplomats". Anwar Sadat is a "president", which is also a "head of state", and fits into many of the same classifications Begin fits into. Information in memory about Begin and Sadat, and the person categories they fit into, is organized as follows:

```
                         heads of state

               "diplomat"
                  E-MOPs: "diplomatic activities" participant
                  /\    /\
                 /        \
                /   "negotiator"
               /      E-MOPs: "negotiations" participant
              /               "diplomatic meeting" participant
             /
     "head of state"
        E-MOPs:  "summit conference" participant
          /\                       /\
          ||                         \
   Menachim Begin                Anwar Sadat
    nationality:  Israeli          nationality: Egypt
    occupation:  head of state     occupation: head of state
    E-MOPs: "Camp David negotiations" participant
```

Figure 5-7

As a head of state, the "summit conference" MOP is associated with Begin. As a "diplomat", "negotiations" and "diplomatic meetings" are associated with him. Begin, himself, is closely associated with "Camp David negotiations", a specialization of the "negotiations" MOP. Consider how the following question can be answered using this information:

(Q5-5) Under what circumstances have you talked to Begin?

Using the information about Begin above, any of the following contexts can be constructed for search in answering this question:

"negotiations about the Camp David Accords with Begin"

"summit conferences about the Camp David Accords including Begin"

"diplomatic meetings about the Camp David Accords including Begin"

The first can be constructed from E-MOP information directly associated with Begin, the second from "head of state" information, and the third from "diplomat" information. Since there is a particular type of "negotiations" context associated with Begin, himself, the general "negotiations" context associated with "diplomats" is not constructed.

Location and object information can be organized similarly. What is important for them, as for contracts and people, is that they be organized in functional hierarchies and that classifications in the

hierarchies specify associated E-MOPs. Thus, with information that "airplanes" have "flying" associated with them, the question "Have you been in a plane recently?" could be answered by constructing and searching for a recent "flying" episode. With E-MOP information associated as above with "resort area", "trips" can be inferred.

5.4 Summary

Generalized information is associated with E-MOPs and other concepts in memory. Because retrieval of events centers on event categories (E-MOPs), and retrieval strategies make use of MOP-related information, it is important that generalized information in an event memory be very rich in referring to events and their categories. Thus, E-MOPs themselves refer to other E-MOPs they are related to, and other memory concepts refer to E-MOPs they are normally associated with. In addition, elaboration requires specification of event components. Thus, E-MOPs must specify what their components normally look like.

Because context construction requires information from concepts other than E-MOPs to infer E-MOPs for traversal, concepts in memory must refer to event contexts they are normally associated with. In fact, automatic context construction in any data base requires that data base concepts refer to the types of valid retrieval contexts they can be associated with.

CHAPTER 6

Automatic Indexing

6.1 Introduction

Index selection is an integral part of both adding a new event to an E-MOP and attempting to retrieve an event from an E-MOP. As new events are added to an E-MOP in memory, they must be indexed in that E-MOP according to their differences from its norms. During retrieval, indices a target event would have if it were in memory are chosen and traversed.

Consider the following event:

EV6-1: Cyrus Vance has a meeting with Andrei Gromyko in Russia about the Afghanistan invasion.

How should his event be indexed in a data base of information about Cyrus Vance so that it can be easily retrieved? Consider the following three cases:

Case 1: Vance has met many times before with Gromyko, but never in Russia, and never about the Afghanistan invasion.

Case 2: Vance has been in Russia for the past two weeks meeting with Gromyko every day about the Afghanistan situation.

Case 3: Vance has never met with Gromyko before, but has met with other foreign ministers before in their country of residence.

Case 4: It is Vance's first week on the job, and he has never been on a diplomatic trip before and never talked to Gromyko.

In the first case, the topic and location of EV6-1 can distinguish it from other meetings in memory. Either of those features would be reasonable indices for EV6-1 in a "diplomatic meetings" MOP. In the second case, however, its location and topic cannot distinguish this meeting from other meetings already indexed in memory. Indexing on those features will not be helpful in discriminating this meeting from others.

There are two major problems to be addressed in discussing index selection:

1. What characterizes a good E-MOP index?

2. How can E-MOP indices for an event be chosen?

In a memory where new categories are created as new items are added to the memory, it is not sufficient only to specify the types of indices initial E-MOPs should have. There must also be (a) a way of judging what would potentially be good indices in newly created E-MOPs, and (b) a procedure for choosing potentially good indices in new E-MOPs based on their parent E-MOPs. Different E-MOPs have different relevant features. The nationalities of participants in a diplomatic meeting can be good indices, but in a social setting, participants' nationalities are not as relevant. Indices appropriate to any particular E-MOP depend on its salient features. Important aspects of an E-MOP can be inferred from what is known to be important about its parent E-MOPs.

6.2 Why index by differences?

Indexing schemes have two major purposes. One purpose of most indexing schemes is to _subdivide categories_ into smaller, more workable parts. If retrieval requires that each member of a category be enumerated during retrieval, then the larger the categories, the slower retrieval will be. While the most efficient way to search a category with a few members might be to check each member individually, when a category gets large, checking individual members is not efficient.

In addition to subdividing a category, indexing can serve to _make members of a category more accessible_. Through cross-indexing, the same item can be referenced in many different ways. Through sub-indexing, an item can be given a unique description. Thus, indexing should make items _discriminable_, differentiating them from one another.

Indexing and sub-indexing should organize memory for efficient retrieval. In a well-indexed well-organized memory, retrieval should not have to slow down as new items are added. As in other large stores of information, indexing in an E-MOP should both (a) divide it into significantly smaller pieces, and (b) provide for discriminability of an E-MOP's events. In that way, retrieval efficiency can be maintained.

We've stated previously that events are organized in E-MOPs according to their differences from the E-MOPs norms or content frame. Why should indexing in E-MOPs be by differences?

Suppose indexing were not only by differences, but also included an E-MOP's norms. Since most events organized in any E-MOP have the E-MOP's norms as features, they would all have virtually the same indices. Thus, indexing by a norm would neither add to discriminability nor divide a memory category into smaller pieces. Instead, it would supply memory with unneeded redundancy, and violate economy of storage.

Differences between events, on the other hand, differentiate them from each other, providing discriminability. Organizing events according to differences allows events to be recognized individually. If a unique difference from a norm is specified in a retrieval key, the event that corresponds to that specification can be retrieved.

Indexing by differences also serves to subdivide an E-MOP into smaller parts. A particular difference from an E-MOP's norm should distinguish only a small number of the E-MOP's items. Thus, only a small number of events in any E-MOP should have the same index and be stored in the same sub-MOP. Each time an index is traversed during retrieval, the number of events that will have to be considered for retrieval will decrease considerably. Thus, by dividing a memory category into smaller parts, an indexing scheme based on differences controls the size of the search space during retrieval.

Imagine, for example, somebody attempting to buy a house. He looks at a lot of houses in one day, and doesn't remember every detail of each one. A few of the houses he sees are yellow. One of those is very large, another small, one in bad shape, etc. Indexing these houses by their color will not aid his retrieval of them for two reasons. First, their color could not distinguish them from each other. All of the yellow houses would be indexed the same way. Second, color is not a relevant feature in house buying. Imagine, however, that one house this person looked at was red with large purple dots painted on it. Would its color be a good index? In this case, the answer is "yes", despite the fact that color is normally irrelevant to house buying.

Discriminability is an important factor in indexing. Because a feature that is different from a norm provides discriminability, indexing an object with a weird color by its color will distinguish it from other similar objects. Indexing on a common color will not do that.

Suppose Cyrus Vance were at a foreign policy meeting with a few other people in a conference room at the White House. Unless there were something special about the room, the meeting's location would not be a good index for it in a meeting E-MOP, even if he had never been in that particular conference room before. It is too much like the locations of other meetings he's been to. Similarly, the number of people at this meeting would not serve as a good index. Because meetings normally happen in conference rooms and have a small number of people, indexing on those aspects would not aid in discriminating this particular meeting from others.

Suppose, however, that the meeting had taken place in Studio 54 (a discotheque). In this case, its location would be a good index in a "diplomatic meetings" MOP, despite the fact that the location probably has no bearing on the meeting or its results. Similarly to the way a red and purple house violates expectations of what houses normally look like, a discotheque violates expectations about the settings of foreign policy meetings. A six-hundred-person foreign policy meeting would also violate normal expectations. Because it violates a norm, the size of the crowd would be a good index for a meeting with an unusually large number of people.

Suppose, however, that a meeting in Studio 54 had to be indexed in an E-MOP which organized the business meetings of the owner of Studio 54. Unlike the E-MOP for foreign policy meetings, the "business meetings involving the owner of Studio 54" MOP might specify Studio 54 as one of its usual meeting locations. In that E-MOP, a meeting in Studio 54 would not be unusual, and its location would not be a good index.

The appropriateness of an index, then, is determined by context. Since indices for events should be differences between an event and an E-MOP norm, index selection in a particular E-MOP depends on the norms the E-MOP has. To decide whether the locations of the two meetings above would be appropriate for indexing the meetings, the location norms of the "diplomatic meetings" and "business meetings of the owner of Studio 54" MOPs had to be consulted.

These examples point out two important indexing rules:

1. Normal aspects of a situation should not be indexed.

2. Weird and different aspects of a situation should be indexed.

In the last chapter, we discussed the information that resides in an E-MOP's content frame. An E-MOP holds generalized information about other E-MOPs it is related to causally and temporally, and also holds generalized information describing its usual role fillers and other components. Now that we have explained the kinds of information held in an E-MOP's content frame, we can explain what a "difference from a norm" is and how they can be chosen during memory update and retrieval.

6.3 What is a difference from a norm?

An aspect of an event can differ from a norm in a number of ways. It can be a variation of the norm, a specialization of the norm, or a generalization of the norm. In addition, extra aspects of an event not mentioned in an E-MOP's content frame can be treated as differences to be indexed.

Thus, we would expect that differences indexed in an E-MOP are differences between aspects of events and corresponding E-MOP content frame information. If a meeting E-MOP specifies that its topics are usually contracts, and if the topic of a particular meeting is "SALT", then the topic can be treated as a specialization of the norm and be indexed. If a meeting is called to plan a summit conference, it too can be indexed by its topic because its topic is different than the norm (i.e., is not a contract).

Cyrus Vance has been to a number of summit conferences. CYRUS' "summit conference" MOP specifies that its topic is usually an international contract, its participants are usually heads of state of the countries involved in that contract, its location is typically a resort area in a country neutral with respect to the contract being discussed, its purpose is usually negotiations, and its usual length is a few days. The content frame of that E-MOP is illustrated below:

"summit conference"

```
content frame:  topic is an international contract
                participants are heads of state of
                  countries involved in contract and
                  a group of their advisors
                location is resort area in neutral country
                purpose is negotiations
                includes many diplomatic meetings
                duration is a few days
```

Figure 6-1

Taking these norms into consideration, how could we expect the "Camp David Summit" to be indexed in that E-MOP? Its purpose was to negotiate an accord between Israel and Egypt, an important international contract. Its participants were the heads of state of Israel, Egypt, and the United States, plus their advisors. Its location was the U. S. president's private resort complex in the United States (not quite a neutral country). It lasted almost two weeks. Figure (6-2) below illustrates a portion of CYRUS' "summit conference" MOP after adding the Camp David Summit and indexing it as just described.

```
                          "summit conference"

content frame: topic is an international contract
               participants are heads of state of countries
                 involved in contract and a group of their
                 advisors
               location is resort area in neutral country
               purpose is negotiations
               includes many diplomatic meetings
               duration is a few days
            /          |               \         \
           /           |               loc       duration
        topic       participants        |           |
          |            |               not        approx
          |         also include U. S. neutral    2 weeks
  peace between     head of state and
  Israel & Egypt   advisors
```

Figure 6-2

Because the topic of the conference is *better specified* than the topic specification of the content frame, this conference is indexed according to its topic. The participants were heads of state of the countries involved in the contract, their advisors, plus the head of state of the United States and his advisors. Because the norm states that participants are usually directly related to the sides of a contract being discussed, the addition of participants from a third arbitrating country are indexed -- a *more general case* than the E-MOP's norm. The conference's location is a resort area of a country with a stake in the agreement being negotiated (i.e., it is not quite neutral). Thus, it is a *difference* from the norm (i.e., a violation) and should be indexed. The purpose of the summit was negotiation of its major topic. This matches the norm, so there is no index for it. Finally, because this summit was longer than the norm (a violation), an index for its length is appropriate.

We shall see in later chapters how processing differs for each of these types of differences as events are added to E-MOPs. New specialized sub-MOPs can be created based on specializations of the norm. Content frame aspects of an E-MOP can be refined and filled in based on violations and generalizations of the norm.

6.4 What makes a good index?

Indexing is used in both adding to and retrieving from memory. In adding events to memory, indices must be chosen to decide where to put new events. In retrieval, choosing indices is necessary for deciding where to search in an E-MOP. Thus, retrievability and discriminability are very closely tied together. The better an index differentiates an event from other events in an E-MOP, the easier retrieval of the event will be. An event will be more retrievable if its indices are as general as possible than if they are very specific.

In order to recognize that something is a good index, we need criteria for judgement. One thing we already know is true about a good E-MOP index is that it must differ from one of the E-MOP's norm. However, not all content frame properties should be indexed to the same level of detail, and the details indexed in two seemingly similar situations might not be similar at all. Consider, for example, the following two situations:

> An employer is interviewing a prospective employee for a job as a computer programmer.

> An employer is interviewing a prospective employee for a job as a fashion model.

What would be reasonable indices for these two events in an employer's "interview" E-MOP? As interviews, there are similarities between the indices which would be reasonable for each. For both of these events, it would be reasonable to index the interview using relevant features of the prospective employees.

The features of the prospective employees that would make reasonable indices, however, differ greatly. It would be reasonable for the employer interviewing a prospective computer programmer to index the interview according to the interviewee's educational background. It would be less reasonable, however, for him to index the experience according to the color of the interviewee's hair. On the other hand, it would be more reasonable for the employer interviewing a prospective model to index the experience according to the person's hair color and what he is wearing, rather than by his education.

An important question that we must answer, then, is how much detail and which details should be indexed for each content frame aspect in any situation? The example above shows that the answer to this question is not absolute, but depends upon the context in which the event is taking place. Thus, in order to answer this question, we must explain the things which regulate how many details and which details of a content frame aspect are relevant for indexing.

6.4.1 Uniqueness

Because indices should divide a category into smaller pieces and differentiate events, unique features of events should make good indices. A unique feature is one for which there is not already an index. If the only diplomatic meeting Vance has ever had with Dobrynin is indexed as a meeting with Dobrynin, then it will be retrievable through that unique feature.

Indexing Principle #1
Unique features of events make good indices.

All unique features of events, however, should not be indexed. Dobrynin is an important political dignitary who Vance might want to talk to again. If he talks to Dobrynin again, it might be useful to remember the first time they met, so that predictions could be made from it, and so that similarities between "meetings with Dobrynin" could be computed. Thus, indexing the meeting above according to who its other participant was makes some sense. Suppose, however, that Vance had a public relations meeting with John Doe, president of the Nuke the Commies Organization. Although it is his first meeting with John Doe, indexing by actual meeting participants would not be the most reasonable participant index in this case.

A more general unique feature of John Doe is that he is a member of a radical anti-Communist organization. If Vance had never met before with a member of a radical anti-Communist organization, the fact that John Doe is a member of this type of organization would be a better index than his name.

Although both his name and his affiliation are unique, John Doe's affiliation is a better index than "John Doe" himself. Suppose Vance had another meeting with somebody from a radical anti-Communist organization. If the first meeting could be retrieved using the features of the second event, then the outcome and events of the earlier meeting could be used to predict the outcome and sequence of events of the later one.

Because "member of a radical anti-Communist organization" is also unique, but more general than "John Doe", it will be applicable as an index for retrieval in more cases. If Vance meets again with John Doe, this index can be traversed to retrieve his last meeting with John Doe. The same index will be traversable if he meets with somebody else from the organization, or if he meets with somebody from a similar organization. In each of these cases, the first meeting with John Doe could be retrieved and used for predictions.

All aspects of unique features of an event need not be indexed, then, but only the most general description of a unique feature. One description is more general than a second if it can be used to describe the second, i.e., if all of its aspects are also aspects of the second. A description of an event feature is unique to an E-MOP if there is no index for it and if it does not describe any other index (i.e., if no other event in the E-MOP has that feature).

In the example above, the two descriptions of the meeting participant -- "John Doe" and "member of a radical anti-communist organization" -- are both unique descriptions of the participant. Because "member of a radical anti-communist organization" describes "John Doe", it is more general and will be a more useful index. We will refer to the following principle, which expresses that, as the "most general unique description" principle.

Indexing Principle #2
The more general a unique feature used as an index, the more retrievable the event being indexed will be.

6.4.2 Predictive power

What if John Doe had been scruffy and had needed a shower? Would those characteristics of John Doe be reasonable ones for indexing the event above in Vance's public relations meetings E-MOP? If predictions about future meetings with scruffy participants could be made from John Doe's appearance, then it would be a good index. If, however, no predictions about meetings could be made from his appearance, then it would not be appropriate.

An important property indices should have is predictive power. A feature which is predictive often co-occurs with some other event feature. The nationality of participants in a diplomatic meeting, for example, is usually the same as one side of the contract being discussed. Thus, in a "diplomatic meetings" MOP, the nationality of participants is usually predictive of another feature, and actual nationalities of participants are good predictive features.

An event feature which is predictive is a feature, which, if it indexed an E-MOP instead of an event, would correlate with other E-MOP features. It is important to index individual events by features which are potentially predictive so that, if additional similar events are added to memory, a reasonable new E-MOP can be formed based on the similarities between events with that feature.

Suppose, for example, that the nationality of participants is known to be predictive for meetings. The first time a meeting with a Russian is added to memory, one way it will be indexed uniquely is by the nationality of its participants -- Russian. The second time a meeting with a Russian is added to memory, it also will be indexed as a meeting with a Russian, and if there are additional features in common, a new E-MOP can be formed and generalizations made about "meetings with Russians".

Predictions that a particular feature or set of features can make are used during retrieval and event reconstruction for elaboration. In reconstructing an event and in elaboration during traversal, specification of the value of a predictive property will allow inference of the properties it predicts. If the feature "participants are Russian" normally co-occurs with "meetings are rowdy", and if

"participants are Russian" is an index in a "meetings" MOP, then knowing that a meeting was with Russians will allow the inference that "the meeting was probably rowdy" during retrieval. If the feature co-occurs with "topic is usually arms limitations", then knowing a meeting was with a Russian will allow inference that the "topic was arms limitations". During traversal/elaboration, that inference can be used to further elaborate aspects of arms limitations that would have been discussed.

Of course, we can't tell for sure, the first time we see a particular feature, whether or not it will later be predictive. The first time a "meeting with a Canadian" is added to memory, for example, we cannot know whether or not "nationality of participants is Canadian" will be a predictive feature. There is a heuristic, however, for judging feature predictiveness. Predictiveness of features can be judged by previous experience. If a <u>type of index</u> (e.g., appearance of participants, nationality of participants, sides of a contract) has been useful previously for similar events, then there is a good chance it will be useful for the current event. Thus, if nationalities of participants has been predictive in indexing other "diplomatic meetings" or "diplomatic activities", then "nationality of participants is Canadian" [illegible]ably be predictive.

Indexing Principle #3

> Indices should have predictive power or potential predictive power.

Thus, as new events are added to memory, the relative predictive power of different types of indices must be tracked. This tracking process will be explained in the next chapter.

In the case of the meeting with John Doe, if no predictions for meetings have previously been made from the appearance of participants, then there is no reason to believe that appearance would be predictive in this case. There would be no reason, then, to index the event in the "public relations meeting" MOP according to the appearance of its participant. Furthermore, even if there are beliefs in the memory that scruffy appearance implies stupidity or uninterestingness or arrogance, "scruffy appearance" would still not be a good index for this particular meeting in the "public relations meeting" MOP. That would be a prediction outside the realm of public relations meetings, i.e., a prediction about personality traits in general, but not about meetings.

E-MOP indices, then, should not only have potential predictive power, but they should make <u>context-related predictions</u>, i.e., predictions about MOP-specific features. They should potentially correlate with some other content frame aspect of the E-MOP the event is being indexed in. Thus, if appearance will not correlate with other features <u>specific to public relations meetings</u>, then it will not have the predictive power to make it a good index for that context.

The fact that appearance is not a good index for participants of public relations meetings does not mean, however, that appearance is a bad index in general. Appearance of participants would probably be a

good index for somebody running a modeling agency and interviewing a prospective model. The appearance of a model is a good indicator of how much money the model will bring into the agency (i.e., outcome), and thus would be predictive in that context.

Whether or not a particular feature will have predictive power, then, depends on the context in which it is found. The usefulness of a feature as an index depends on the E-MOP it is being used in. Good E-MOP indices, then, should not only be potentially predictive, but should predict MOP-specific aspects of later events.

Indexing Principle #4
The usefulness of an index depends on its context; a good index should make context-related predictions.

The heuristics presented for judging potential predictiveness of features is not fool-proof. A feature that is judged to be a good index at one time might sometime later be judged not to be useful. "Nationalities of participants" might be predictive in general, but the particular index "nationality of participants is Canadian" might not be predictive of other features. Similarly, a nice appearance might turn out not to correlate with any other event-related features, while a scruffy appearance does correlate. After additional events are added to memory and it is discovered that a particular index does not correlate with other event features, the bad index should be marked as no longer useful, and not be used anymore. Thus, the feature "nice appearance", though it might not be a norm of an E-MOP, could be marked as not useful for indexing, while the feature "scruffy appearance" might still be a good index for that E-MOP. In the next chapter, the process of refining the knowledge used for indexing will be explained.

6.5 Choosing indices

In the last section, we questioned how much detail and which attributes of content frame aspects should be indexed. In beginning to answer this question, we defined criteria for judging whether an index is good. A good E-MOP index has predictive power, is contextually-related, and is different from the E-MOP's norm. We can now judge potential indices according to these criteria and determine which particular indices are good in particular contextual domains.

Criterial judgements on the usefulness of indices is only one step on the way to answering this question. We must also explain the knowledge necessary for index selection and the process that uses that knowledge. In this section, we will do this.

Since only features with predictive power should be indexed, the actual process of choosing indices must first use information about predictability to select possible indices for an event. Each of those possibilities can then be checked to make sure (1) that it has not been marked as non-predictive, and (2) that it is not an E-MOP norm. In addition, each feature chosen as a potential index must be checked to

see if there is a unique way of describing it. If so, the most general unique way of describing it should be used as an index.

Index selection

```
+----------------------------------+
|  Select types of features from   |
|  event that have been predictive |
|  previously                      |
+----------------------------------+
                 |
                \ /
  +--------------------------------+
  | Get rid of                     |
  |  (1) particular features known |
  |      to be non-predictive      |
  |  (2) E-MOP norms               |
  +--------------------------------+
                 |
                \ /
+--------------------------------------+
| Choose most general way of describing |
| those features which are unique       |
+--------------------------------------+
```

Figure 6-3

In order for predictive power to be used in choosing indices, each E-MOP must keep track of types of indices which have been useful, and actual features which have not been useful and should no longer be used as indices. Thus, if "nationalities" have been useful previously in indexing "meetings", then the nationality of participants in a particular meeting will be chosen in the first step to index that meeting in a "meetings" MOP. If the particular nationality of the meeting's participants has been marked as non-predictive, or if it is a norm, then it would be deleted from the set of possible indices in the second step.

Consider, for example, CYRUS' "diplomatic meetings" MOP after enough meetings have been added to it for it to know some of its non-predictive features. The following are some of its predictive aspects, its norms, and the features it knows to be non-predictive at that point in time.

"diplomatic meetings"

content frame:
  participants: foreign dignitaries of countries involved in contract being discussed (i.e., diplomats)
  location: conference room in capital city of country of residence of some important participant
  topic: international contract
  duration: one to two hours

predictive:
- political roles of the participants
- classes those roles fit into
- nationalities of the participants
- occupations of the participants
- political leanings of participants
- topic
- sides of the topic
- issue underlying the topic (e.g., peace)

non-predictive features:
- participants' occupation is for. min.
- participants' occupation is head of state

Figure 6-4

Suppose the meeting with Gromyko presented at the beginning of this chapter (repeated below) were to be indexed in this E-MOP.

EV1: Cyrus Vance has a meeting with Andrei Gromyko in Russia about the Afghanistan invasion.

Using predictive aspect information in the E-MOP above, plus the knowledge that in general the place and participants in an activity are predictive, the following features would be chosen for indexing in the first step of the index selection process.

1. the meeting is with a foreign minister
2. the meeting is with a diplomat
3. the meeting is with a Russian
4. the meeting is with a Communist
5. the meeting is about SALT
6. the topic concerns the U.S. and Russia
7. the issue underlying the topic is arms limitations
8. the meeting is with Gromyko
9. the meeting is in Russia

In the second step of the process, those features known to be non-predictive or norms of the E-MOP are deleted from the set of potential indices. Although in general, people's occupations can be predictive of other meeting aspects, the particular occupation "foreign minister" is not predictive for diplomatic meetings. Thus, feature (1) is deleted from the set of indices. Since participants in "diplomatic meetings" are normally diplomats, feature (2) is deleted from the set. Taking into account the norms and non-predictive aspects of the "diplomatic meetings" MOP above, the following features would remain as plausible indices for EV1 after step 2.

A. the meeting is with a Russian
B. the meeting is with a Communist
C. the meeting is about SALT
D. the topic concerns the U.S. and Russia
E. the issue underlying the topic is arms limitations
F. the meeting is with Gromyko
G. the meeting is in Russia

One potential problem with indexing events in many different ways is combinatorial explosion of indices in memory. This problem can be controlled, however, by indexing only differences from an E-MOP's norms which are predictive of other features. In that way, indexing at lower levels of the memory structure is constrained. Knowledge about an E-MOP's norms, and about which of its features are predictive, then, is necessary in order to limit indexing. Keeping track of norms and non-predictive features is an important aspect of integrating new events into memory. The next chapter will discuss the process of integrating a new event into memory. In discussing that process, the building up of norms in newly created E-MOPs and their control of subsequent indexing and new E-MOP creation will be explained.

6.6 Classifying E-MOPs to guide indexing

Many of the features specified above for "diplomatic meetings" are true not only of meetings, but also of other classes of activities meetings fall into. "Diplomatic meetings", for example, are communicative activities, political, and occupational. It seems likely that the topic of any communicative activity would be predictive of other aspects of the situation. Similarly, it seems likely that for any political activity, the polities involved in the activity would make good indices, and for any occupational activity, the occupations of other participants and the occupational role being played in an activity could be predictive.

The "diplomatic meetings" MOP, then, should not have to keep track of all indexing information for meetings. As a "communicative activity", "diplomatic meetings" share features with conferences, discussions, and consultations. The topic of any "communicative activity", for example, is probably appropriate as an index. If "communicative activity" specified its norms and which of its aspects were potentially good index types, then that information could be used

by the "diplomatic meetings" MOP, the "conference" MOP, etc., and would not have to be stored redundantly on each particular communicative MOP.

Classifying E-MOPs according to more general E-MOPs and E-MOP categories they are specializations of has two advantages. First, E-MOPs and E-MOP classifications for which the nature of good indices was already known could be used to guide choice of indices in more specialized E-MOPs. If the general nature of good indices for political events is known, for example, it can be used in deciding how to index events in "diplomatic meeting", "diplomatic trip", and other political E-MOPs. Second, generalized information common to a group of E-MOPs could be stored non-redundantly in one place, instead of on several different E-MOPs.

An important implication of this is that as new E-MOPs are created, knowledge about predictive features can be inherited from the E-MOPs it is created from, and knowledge about which features are appropriate for indexing does not have to be built up from scratch.

E-MOPs can be classified in "E-MOP classes" similarly to the way events are classified by E-MOPs. These categories should be, as E-MOPs are for events, memory structures which <u>organize the similarities between E-MOPs</u>. Formal "political activities", for example, always involve discussion of some political topic and always involve political dignitaries interested or involved in the topic of discussion.

"Occupational activities" are activities done for employment purposes, and involve people related through employment. A low-level person does things because his higher-up asked him to or expects him to. In addition, occupational activities generally relate to other domains. While Vance was Secretary of State, for example, almost all of his occupational activities were also political.

In CYRUS, the major E-MOP classifications are "international political activities", "occupational activities", and "social activities". Most of the things Vance does which CYRUS knows about are related to his occupation, and are both occupational and political. Some of the things he does, such as parties and state dinners, are also social.

Each classification groups together a packet of information common to a number of different E-MOPs. Thus, each E-MOP classification holds the same types of information E-MOPs hold. The major difference between an E-MOP and an E-MOP classification is that E-MOPs organize both memories and other E-MOPs, while an E-MOP classification organizes only other E-MOPs.

"Political activities", as a classification, keeps track of a large number of generalizations true of all political events. All political activities, for example, have as participants people with political occupations (i.e., political dignitaries), and the political affiliation of participants in a political event can be predictive of the topic of discussion. Their topics can be predictive, since they affect international relations. The topic of discussion of a political activity has sides, and representatives of each side take part in the

discussion. Thus, the sides of a topic being discussed can be predictive. The topic usually has an underlying issue such as peace, arms control, aid of some sort, or international relations, which is also predictive of future international relations. These aspects of any political event are appropriate for indexing.

In addition to references to "political activities" and other classifications an E-MOP fits into, individual political E-MOPs need only specify their own MOP-specific information. In general, an E-MOP need only specify the E-MOP classifications and E-MOPs it is specialized from, plus its own MOP-specific information (i.e., differences between its own features and the feature descriptions in the E-MOP classes it belongs to). It can inherit the information stored in the classifications it belongs to.

Because the "diplomatic meetings" MOP in CYRUS is political and occupational, and a type of meeting, the types of features it indexes are chosen from among the types of indices marked as predictive for those types of activities. CYRUS chose the following indices for a diplomatic meeting between Vance and Dobrynin in Washington in which they discussed the MIG question.

1. participant was Dobrynin
2. nationality of participant was Russian
3. occupation of participant is Ambassador
4. topic was the MIG question
5. topic related to military issues
6. sides of the contract discussed were Russia and the US

Because it is an "international political activity", its topic (4), the countries it involved (6), the issues underlying the topic (5), and the nationality (political leaning) of the participants (2) are indexed. Because it is an "occupational activity", the occupations of other participants are indexed (3). The event is indexed by its participant (1) because CYRUS has never seen a meeting with Dobrynin before (i.e., the meeting is unique in that way), and because CYRUS always indexes events by their participants.

Notice that the location of the meeting does not get indexed. There are two reasons for this. First, most diplomatic meetings occur in Washington. Thus, it is the norm. Second, if Washington were not the norm, the location would be indexed only if (a) meetings had happened there only a few times (i.e., if it was not known yet if "Washington" would be predictive), or (b) if meetings had happened there many times and there was some other meeting-specific feature that they had in in common (i.e., if "location = Washington" were predictive of other features).

The following three sections explain CYRUS' context-based E-MOP classifications in detail and show how these classifications are used in CYRUS to choose indices.

6.6.1 Political activities

CYRUS uses the classification "international political activity" to classify those E-MOPs in which people of other nationalities play a large role, or events which always take place in a foreign country. Thus, "diplomatic meeting", "diplomatic trip", "summit conference", and "negotiations" are all international political activities. Each of these E-MOPs has information peculiar to itself and inherits a packet of information shared by all international political activities. That packet of information includes descriptive features of "international political activities" plus indexing information -- knowledge which can be used for index selection.

Politically-oriented activities can be differentiated, discriminated, or indexed by the political entities involved -- either as participants, locations, or topics of discussion. Reasonable indices for political events include the following:

1. the political entities involved as participants
2. the political roles of the participants
3. the political relationship between the participants
4. the political aspects of the topic of discussion
5. the politics of the location of the activity
6. the appropriate political leanings of the participants
7. other related political events going on around this one

How do we know these are reasonable indices for political events? These are reasonable if they can contribute to good predictability and discriminability. The topic of discussion during a political activity is usually an issue of importance to the political entities involved. Thus, the political entities involved as participants in a political event can help predict its topic of discussion. The political role of a participant may predict his behavior and visibility. A negotiator or arbitrator or a chief delegate will act differently in a political activity than will a lower level delegate.

Particular types of political activities can better constrain those indices. In "international political activities", it is international political entities which are predictive. Thus, the nationality of participants, their international political leaning (e.g., capitalist vs. communist), the area of the world they are from (e.g., eastern vs. western, Middle East, South American), and their political roles might all be good indices. During traversal/elaboration, each can predict another aspect of a meeting -- nationality predicts topic, while other aspects predict meeting outcome, sides in an argument, etc. A description of the location of an activity with respect to its topic and participants can also be a good index -- is it a politically neutral country, is it in a country with the same leaning as the participants, is it in a country with the same leaning as one side of a contract being discussed? If more concessions are made by a country hosting a meeting, then the meeting location will be a very good index. If the meeting is held in a neutral country, that would predict the explosiveness of the topic and make a good index by the criteria above.

CYRUS initially knows that the participants of international political activities have political occupations, are diplomats, and play political roles in these activities. It also knows that the topics of international political activities are usually international contracts, that political events are included in other political events, that they include other political events, and that they are often initiated by an international conflict.

CYRUS also knows that the political roles participants play in political activities (e.g., negotiation, arbitrator) might be predictive of other information, and therefore good indices, and also that the classes those rules fit into (e.g., diplomat, bureaucrat), the nationalities of participants, the countries involved in any contract being discussed and the political leanings of people and countries involved can be predictive indices. The following is CYRUS' knowledge about "international political activities". Each E-MOP classification has the same types of knowledge associated with it that E-MOPs have.

"International Political Activities"

content frame:
- participants' occupations are political
- participants are diplomats
- topic is an international contract
- included in other "international political activities"
- includes other "international political activities"
- initiated by international conflict

predictive specs:
- political roles of the participants (e.g., negotiator)
- classes those roles fit into (e.g., diplomat, bureaucrat)
- nationalities of the participants
- polities on the sides of any dispute or contract being discussed
- political leanings of participants

Figure 6-5

The predictive aspects of "international political activities" are used by CYRUS to choose features of an event as possible indices in the E-MOP it is being added to. The generalized information is used to narrow down the possible indices. Those features which are the same as content frame features will not be indexed.

Suppose, for example, that we wanted CYRUS to choose possible indices for the following event.

(EV2) Cyrus Vance meets with Andrei Gromyko about SALT in Belgium.

Because this event describes a diplomatic meeting, CYRUS uses knowledge from its "diplomatic meetings" MOP to choose indices. A "diplomatic meeting" is a "meeting", an "international political activity", and an "occupational activity". If Vance is involved in it, it is also a "Vance activity" (any activity Vance is involved in). "Vance activities" have as their norms that Vance is a participant and that they take place in Washington, in that way helping to limit indexing. All of these things must be taken into account in choosing indices.

CYRUS knows that meetings, in general, are a few hours long and are about contracts. It also knows that the underlying issue of the contract being discussed can be predictive, as can the sides of the contract, and the class the sides of the contract fall into. Thus, CYRUS uses its knowledge about meetings to choose the following features as possible indices for this event in the "diplomatic meetings" MOP.

(1) The contract is SALT
(2) The contract is a military contract
(3) The contract is international (i.e., between countries)
(4) The contract is between the United States and Russia

Besides being "meetings", however, "diplomatic meetings" are also "international political activities". Using only meeting information, the indices CYRUS can choose are incomplete. Because international political activities usually have an international contract as their topic, that feature should not be used as an index. In addition, some potentially good indices have not been chosen. The fact that the participant was Russian or Communist cannot be chosen as an index using only meeting information. Meeting specifications, by themselves, do not specify that participants' political leanings can be predictive.

Using the "international political activity" information illustrated above in figure (6-5), a more complete set of indices for the meeting can be chosen. The fact that the topic is an international contract would be deleted from the set of possible indices. Using the information that participant's nationalities and political leanings, the role of the participants, and the polities involved in a dispute can be predictive for international political activities, the following indices can be chosen in addition to the ones above:

(a) Participant is Russian
(b) Participant is Communist
(c) Participant is a diplomat
(d) Participant is a negotiator

Taking the norms of "diplomatic meeting" into account, however, the roles of the participants (c, d) would be deleted from the set of possible indices in this case, leaving the following:

(1) The contract is SALT
(2) The contract is a military contract
(3) The contract is between the United States and Russia
(4) Participant is Russian
(5) Participant is Communist

This list of indices, however, is still not complete for event (EV2), since it takes only "meeting" and "political activity" aspects of "diplomatic meetings" into account. "Diplomatic meetings" are also "occupational activities" and "Vance activities", and features of each of those must also be taken into account in indexing. As "Vance activities" (all E-MOPs in CYRUS' Vance data base are "Vance activities"), the participants and location of the event should be indexed. CYRUS always indexes events by their participants and location unless a particular index or location is marked as non-predictive. Thus, the following three indices would also be chosen for this event.

(6) Gromyko is a participant.
(7) The meeting is in Belgium.
(8) The meeting is in Europe.

In the next section, "occupational activities" will be described, and choice of the remaining occupation oriented indices for (EV2) will be shown.

6.6.2 Occupational activities

Any event associated with the actor's occupation is an occupational event. The fact that an event is occupation-related means that it might have some employment-centered attributes which would make reasonable indices. The role of the actor within his occupational role may be a good index for occupational events, as will the occupations and occupational roles of the other participants. A professor, for example, acts as teacher, advisor, researcher, and committee member. His role during any activity determines his concerns and his relationship with other participants. Thus, if his situational role in an activity is not a norm, it will make a good index. Similarly, the roles and occupations of other people he interacts with in his job-related roles may make good indices. A meeting with the university president may be like a meeting with the department chairman in many ways, but the fact that it is the university president will add extra social constraints and considerations to a situation. Also related to the roles a person plays in this occupational role, the employment relationship between participants in an event can be a good index. In occupational activities, the relationship of the event toward the progress of some job-related goal may also be a good index.

CYRUS uses the classification "occupational activity" for all of its E-MOPs that are usually done as part of the secretary of state's job. All of CYRUS' political E-MOPs are also occupational. In addition, "speeches", "state dinners", "consultations" and "briefings"

are all occupational activities.

CYRUS knows that the participants of occupational activities are in job-related roles when they are participating in an occupational activity. It also knows that a participant's role in an activity could be predictive of other information, and therefore would be good indices. The following is CYRUS' knowledge about "occupational activities".

"Occupational Activities"

content frame:
- participants are in occupational roles
- included in other "occupational activities"
- includes other "occupational activities"

predictive specs:
- occupations of the participants
- roles of the participants in the event
- classes those roles fit into
  - (e.g., diplomat, bureaucrat)

Figure 6-6

CYRUS uses this information to choose indices in the same way it uses its information about "political activities". It uses predictive aspects of "occupational activities" to choose possible features for indexing, and its content frame information (i.e., norms) is used to decide whether or not a possible feature should actually be used as an index.

In the previous section, we showed index selection for a meeting between Vance and Gromyko (EV2, repeated below) using "international political activity" and "meeting" information.

(EV2) Cyrus Vance meets with Andrei Gromyko about SALT in Belgium.

Occupational aspects of the meeting must also be taken into account in choosing its indices. In choosing indices for this meeting, the predictive aspects specified for occupational activities can be used to choose the following two features as possible indices for (EV2) in the "diplomatic meetings" MOP.

(a) The actor is acting as negotiator.
(b) The actor is acting as a diplomat.
(c) The participant is a foreign minister.

Because the actor (Vance) in Vance's "diplomatic meetings" MOP normally acts as negotiator and diplomat, (a) and (b) must be deleted from the set of possible indices. The fact that the other participant is a foreign minister (c), however, remains to be indexed. Thus, using "meeting", "political activity" and "occupational activity", the following nine indices for the meeting with Gromyko in Belgium would be chosen:

(1) The contract is SALT
(2) The contract is a military contract
(3) The contract is between the United States and Russia
(4) Participant is Russian
(5) Participant is Communist
(6) Gromyko is a participant.
(7) The meeting is in Belgium.
(8) The meeting is in Europe.
(9) The participant is a foreign minister.

Occupations normally have some relation to some other aspect of one's life. Because of that, particular occupations and occupational events can take on predictive aspects of those other contextual domains. Most of CYRUS' occupational E-MOPs, for example, are also political E-MOPs, and aspects of both are taken into account in indexing. Any indices for events that could be provided through looking at them as political activities would also be good occupational indices. Vance's work as Secretary of State involved him in international political activities. Thus, political aspects cannot be separated from occupational aspects of his activities in many cases.

Good occupational indices, then, must take into account attributes of an actor's occupation. A particular occupation must contribute its norms and information to produce good predictive and discriminating indices. Thus, good indices for Vance's activities as a lawyer would include his various roles as a lawyer, the kinds of relationships lawyers have with clients and other lawyers, and case-related features. Similarly, occupational activities in the life of a minister should be indexed according to both occupational and religious aspects of his job.

6.6.3 Social occasions

Like political and occupational activities, "social activities" have a number of predictive aspects appropriate for indexing. In social activities, the social relationship of the participants will be appropriate for indexing. So will the social group (e.g., which clique) the participants belong to and the social organization, if any, the participants are affiliated with. In real life, personality characteristics of the participants in a particular social setting may also be good predictive indices, since it seems likely that people monitor whether or not they would like to do any activity again with the people or groups of people involved.

In CYRUS' domain, social occasions are all occupational activities as well. Social occasions CYRUS knows about are "state dinners", "dining", and "parties". CYRUS knows that social activities often have hosts, and *that it is important* to index those activities according to who the host was. If CYRUS were to be used to make plans, this would be in keeping with social protocol -- it is appropriate to reciprocate on social invitations.

The following is the generalized information CYRUS has about "social activities":

"Social activities"

content frame: participants include a
host -- the inviter
guests -- the invited

predictive specs:
the host
the other participants
the relationship of the host to the actor

Figure 6-7

As for the other E-MOP classifications, "social activities" contains predictive and normative information which CYRUS uses to choose indices for social events. Consider, for example, the following event:

(EV3) Vance has dinner with Moshe Dayan and Ezer Weizmann in Israel. Weizmann is the host.

Using "social activities" information, the following features are possible indices for this event in the "dining" MOP:

(1) Weizmann is the host.
(2) Dayan is a participant.
(3) The host is related through employment.

Because this is also an occupational and political activity (since the host is related through political employment), indices for political and occupational activities must also be chosen. Using those specifications, plus knowledge about "Vance activities" in general, CYRUS chooses the following additional indices for this event.

(4) The host is Israeli.
(5) The host is a foreign minister.
(6) The other participants are Israeli.
(7) The dinner took place in Israel.
(8) The dinner took place in the Middle East.

The dinner is indexed in the "dining" MOP ($DINE) using these eight features as indices.

6.7 Automatic indexing and information retrieval

How do these indexing methods relate to indexing methods in information retrieval? The goals of indexing in information retrieval are the same as our goals in designing an intelligent fact retrieval system. Memory categories should not have to be enumerated during search. Rather, as has been previously stated, indices should sub-divide a category into smaller pieces and should make members of a category more accessible. Indexing methods should be general enough to provide good recall, but specific enough so that not too many irrelevant items are retrieved (Borko & Bernier, 1978, Salton, 1975).

Although the goals of indexing in information retrieval and in CYRUS are the same, the actual practice is quite different. The reason for that difference is that information retrieval systems index by key word only. Their methods of providing good recall include synonym lists, word truncation, substitution of universal symbols, and associative indexing. All are key word methods and the data they use is collected by statistical processing of the items in the data base.

In CYRUS, on the other hand, indices are based on salient features of a conceptual category. They are knowledge based, rather than statistically based, as in IR systems. The goal of good recall (i.e., accessibility) is provided by defining salient features for a category and indexing on those features. The goal of specificity, or not retrieving irrelevant items, is provided by the elaboration processes, which use knowledge about conceptual categories and knowledge-based correspondences of components to infer relevant indices.

6.8 Summary of indexing

The process for choosing indices uses predictive information on E-MOPs and E-MOP classes to guide its selection. When a feature to be indexed is chosen because of its possible predictive power, it is then checked to make sure (1) that it is not an event norm, and (2) that it has not been marked as non-predictive. The feature should also be checked to see if it has a unique description, and if so, its most general unique description should be indexed.

Thus, for any content frame aspect being indexed, its level of detail will depend on how similar other indices and other fillers of the same aspect are. The first time a person dines out during a job interview, that dining experience can be indexed as having been during a job interview. After he has dined out during job interviews a number of times, however, indices for larger episodes the event was part of will have to be more detailed, and will start to look like "during an interview at a small company" or "during an interview in Atlanta". After many such experiences, we might expect more detailed indices such as "during an interview at IBM in Atlanta".

Detail does not have to be added to indices forever, however. Non-predictive features of activities should not be indexed. Thus, information about predictive power must be maintained. When an aspect of a potential index fails to have predictive power, that aspect of the description of the feature being indexed should no longer be indexed or included in indices. Thus, if after many experiences dining out during job interviews, the size of the company does not predict anything about the meal, the size of the company will not be used in indexing larger episodes the meal was part of. Thus, indices such as "during an interview at a small (large, medium sized, ...) company" will no longer be made.

Predictive power is an important criteria for choosing indices. Since any feature marked as non-predictive will not be indexed, predictive power, along with norms, helps limit the number of indices and level of detail that has to go into an index. Maintenance of predictive power information will be discussed in the next chapter.

E-MOPs can be classified in contextually-related E-MOP categories which can be used to help determine which features of content frame aspects might have predictive power. For international political events, that includes the sides of an international conflict and the political leanings of the sides. For occupational activities, it includes the occupational roles being played by the actor, occupational roles played by other participants, and relationships between those roles. Classifying E-MOPs in categories such as these constrains the number of features relevant for indexing. In political activities, politically-significant attributes of role fillers make good indices. Non-political attributes, such as hair color of the participants, do not have to be considered for indexing except when they are relevant to a particular political E-MOP.

CHAPTER 7

Generalization and Memory Reorganization

7.1 Introduction

As new events are added to memory, its organization must be maintained. Because it is impossible to anticipate beforehand the events that will be added to memory, memory's organization must be able to change to accommodate new, unexpected events. A current events data base whose categories are tuned to today's current events and issues, for example, must be able to integrate new events into its memory. Although unexpected, a memory which knows about "war", for example, should be able to accommodate a sudden brushfire war in the Middle East. In this chapter, the process of integrating a new event into memory will be explained.

7.2 The issues

Integration of a new event into memory means "understanding" the event. In both memory update and understanding, appropriate points in memory from which to draw inferences and make assumptions must be found. Understanding an event includes noticing the similarities and differences between the new event and events already in memory. Integrating a new event into memory must therefore also include noticing those similarities and differences. In understanding an event, previously made generalizations can be used to infer aspects not explicit in the event and to make predictions about what will be true in the near future. Imagine, for example, hearing about the following event:

EV7-1: Cyrus Vance went on a diplomatic trip to the Middle East.

In understanding this event, one would assume that Vance flew to the Middle East, and that he went there to negotiate. A person unaware of current political problems in the Middle East might reply after hearing about this event, "What was Vance negotiating there?" In order to ask

this question, the person had to know that "diplomats generally go on diplomatic trips to negotiate". In other words, he has to use generalized information available in memory.

Where does this generalized information come from? This piece of information looks like information from a "diplomatic trips" MOP. In order for generalized information about events to be available for understanding, it must be built up as new events are added to memory.

In the normal course of understanding people notice similarities between a new event and previous events in memory. As people notice similarities, they make generalizations, and use those generalizations for later understanding.

It would be reasonable to imagine a person who has been to a number of faculty meetings, all of which were boring to him, to make the generalization "faculty meetings are boring", and to add that information to his knowledge about faculty meetings. In doing that, he would be understanding that this faculty meeting is similar to previous ones (in that it is boring), and also updating his information about faculty meetings.

The features of individual episodes provide a framework for deciding whether two episodes are similar to each other, and for making generalizations. If, on two different trips that Cyrus Vance takes to the Mid East, he is welcomed at the airport and then has a meeting with the president of the country he is in, then the sequence of events of the two trips will look alike. We would expect them to be indexed in the same E-MOP. When the second trip is entered into memory, it should *remind* (Schank, 1980) memory of the first one because of their similarities. In the same way, a person hearing about the two trips might be reminded of the first when hearing about the second.

If the result of the first trip had been a signed accord, the person may predict, or at least hope, that an accord would be signed at the end of this trip also. If an accord is signed (i.e., if the results of the two trips are the same), a generalization can be made that "when the first event of a diplomatic trip is a meeting with the head of state, an accord will be reached". That generalization can be stored on the content frame of the "diplomatic trips" MOP which organizes those two trips. Next time a relevant trip is processed, the generalization can be accessed.

Noticing similarities between events, then, is an important part of understanding a new event and integrating it into memory. People, however, do not notice all similarities and differences. A person going to a meeting, for example, is not likely to connect the weather with the meeting unless it has some bearing on the meeting (e.g., how hard it is to get there). The differences and similarities that people notice are those directly related to the relevant episodic context.

As new events are added to E-MOPs in CYRUS, they are indexed in those E-MOPs. Indexing in E-MOPs serves to both discriminate individual events and divide an E-MOP into reasonably-sized subcategories. When two events are indexed in the same way, a sub-MOP of the parent E-MOP is

formed, indexed in the parent E-MOP by the common feature that indexed the two events. CYRUS extracts the similarities between the two events, and stores them in the new E-MOP's content frame to use later in guiding retrieval and in limiting indexing.

An E-MOP's content frame, then, is not static, but changes with the addition of new events. As new experiences are indexed in an E-MOP, its content frame can change. Because later information might contradict some previous generalization, and because all new E-MOPs which are created might not be helpful, memory update must provide for recovery from both bad generalizations and useless E-MOPs.

In describing retrieval and memory search in previous chapters, we have been assuming that memory would be efficiently organized. Because all retrieval specifications and new entries to memory cannot be anticipated, there is no organization of memory which will enable easy memory access every time. A good organization, however, will insure that only a small number of sub-MOPs will have to be searched at any time.

There are a number of problems that must be addressed in describing a process for updating memory and maintaining good memory organization. Some have been introduced above. Some have been discussed in previous chapters. The following list summarizes the issues.

1. how can good memory organization be maintained?

2. how are new E-MOPs created?

3. how can the content frame of an E-MOP be filled in?

4. how is the correct E-MOP for updating chosen?

5. how is a new item entered in an E-MOP?

6. how do the E-MOPs relate to each other?

7. when is it appropriate to create a new E-MOP?

8. what is the role of generalization in memory update?

9. how can generalization be directed?

10. how can we recover from useless E-MOPs and generalizations?

This chapter will address each of these problems except for the sixth, which has been discussed in chapter 5.

7.3 Adding a new event to an E-MOP

Adding an event to an E-MOP means indexing the event correctly in the E-MOP. The first step in adding a new event to an E-MOP is extraction of its MOP-relevant features. This process, the process of index selection, was explained in the last chapter. The actual processing when an event is added to an E-MOP depends on the relationship of each of the selected features to events already indexed in the E-MOP. One of the following three things will always be true about each feature of an event specification:

1. It is unique to this event in the E-MOP, i.e., there is no index for it already

2. It is semi-unique to this event in the E-MOP (it has happened once or a small number of times before)

3. It is often true of events in the E-MOP

The first of these relationships is easy to handle. If a feature has not been indexed before (1), an index is built for it. This event will then be indexed uniquely in that E-MOP. The second and third cases are more interesting. They trigger generalization processes which build up content frame information. Because the process of choosing features for indexing gets rid of those features which match the E-MOP's norms, it will never be true that a selected feature for indexing will match an E-MOP's norm. Thus, *these three relationships between selected indexing* features and E-MOP indices are the only possible relationships.

7.3.1 Initial indexing

In case 1, when the descriptive property is unique, the event is indexed under that aspect in the E-MOP. Any time an event is indexed by a feature unique to an E-MOP, it can be retrieved from that E-MOP by specifying that feature. Thus, the more features an event has that are unique to an E-MOP, the more ways there will be of retrieving it uniquely. The following rule summarizes the process of creating a new index:

Index Creation

IF there is no prior index for a relevant feature of an event

THEN

(1) construct an index
(2) index the event's description there

Figure 7-1

One of the discriminations CYRUS makes on meetings is the underlying topic of the contract being discussed. A meeting about the Camp David Accords is indexed as a meeting whose underlying topic is "peace", and a meeting about military aid to Pakistan is indexed as a meeting with underlying topic "military aid". The first time CYRUS hears about a meeting in which Vance discusses military aid, it will index that meeting uniquely in the "diplomatic meetings" MOP under the property "underlying topic = military aid". From past experience, it knows that the underlying topic can be predictive of other event aspects in political events. Another aspect of "diplomatic meetings" that CYRUS knows can be predictive is the occupation of the participants. Thus, if this meeting were also the first meeting Vance had had with a defense minister, then CYRUS would also index it uniquely under the occupation of its participants. Later, in answering questions, if a meeting's unique aspect is mentioned in a question, it will be directly retrievable without elaboration.

Suppose the memory organization of CYRUS' meetings looked like the following before adding any meetings about military aid to memory.

```
                    "diplomatic meetings" -- $MEET

     content frame:  included in "negotiations"
                     participants are foreign diplomats
                     topic is an international contract
                     is political and occupational for Vance
                        /         |           |        \
     diffs:            /          |           |         \
          underlying              |     participants     |
            topic                 |     nationality      |
           /  |  \                |       /     \        |
       peace  |  trade            |     Israel  Arab     |
      (MOP3)  |  (EV2)            |    (MOP6)  (MOP7)    |
          disarmament             |                      |
             (MOP4)          participants            included in
                             occupations                 |
                              /   |   \                 summit
                             / prim min \             conference
                     for min  (MOP9)    def min        (MOP8)
                     (MOP5)             (MOP10)
```

Figure 7-2

In this "diplomatic meetings" MOP, there are indices for the underlying topic of the contracts being discussed, the occupations and nationalities of the participants, and the larger episodes the meeting is included in. Each index indexes either an E-MOP (labeled MOPn) or an event description (labeled EVn). Suppose a meeting between Vance and the defense minister of Pakistan about military aid were added to this E-MOP. The event description of the event would include the information that the meeting was with a defense minister, a Pakistani, and that its topic was military aid -- its differences from the E-MOP's norms. If we call this event description EV7 and add it to the E-MOP illustrated above, that E-MOP would have the following structure:

```
                 "diplomatic meetings" -- $MEET

 content frame:  included in "negotiations"
                 participants are foreign diplomats
                 topic is an international contract
                 is political and occupational for Vance
                 /           |           |          \
 diffs:         /            |           |           \
        underlying           |     participants       |
         topic  \            |     nationality        |
        /   |  \  \          |       /    | \         |
 peace      |   \   trade    |     Israel | Arab      |
 (MOP3)     |    \  (EV2)    |     (MOP6) | (MOP7)    |
     disarmament  \          |            |           |
      (MOP4)   mil aid       |       Pakistan         |
                 (EV7)       |         (EV7)          |
                     participants             included in
                     occupations                   |
                     /     |     \               summit
                    /  prim min   \            conference
              for min  (MOP9)    def min         (MOP8)
              (MOP5)             (MOP10)
```

Figure 7-3

Two new indices, one for the underlying topic of the meeting, and one for the nationality of the participants would have been added to the diplomatic meetings MOP, and this new event, EV7, would be indexed uniquely at those points. In addition, it would have been added to the E-MOP "diplomatic meetings with defense ministers" (MOP10 above, not shown in detail), and indexed in that E-MOP according to its differences from that E-MOP's content frame. Because there is only one "diplomatic meeting about military aid" in memory at this point, if we asked CYRUS to recall a "diplomatic meeting about military aid", it would be able to retrieve EV7 without the use of retrieval strategies.

7.3.2 E-MOP creation

If a property being indexed has occurred only once before in an E-MOP (case 2), we say *reminding* (Schank, 1980) has occurred. When two events have the same set of features, they will be indexed in the same way in an E-MOP. When the second event is indexed at a point where there is just one other event, the previous event is remembered. This is reminding. Suppose, for example, that one of Vance's meetings had included a meal at a very fancy restaurant, and that it was his first working lunch that had been at a restaurant. A second working lunch at a restaurant would be indexed in memory in the same way as the first one. Upon indexing the second one, the first one would be found.

Reminding triggers the creation of new E-MOPs. When reminding occurs, the current and previous events are compared for common aspects. Similarities between the two events are extracted, and a new E-MOP with generalized information based on those two occurrences is created. In the example above, a new E-MOP "working lunches in restaurants" would be created from the two events, and their similarities would make up its content frame. Features compared for similarity are chosen by applying the index selection process. E-MOPs are created as follows:

E-MOP Creation

IF there is one other event indexed at an index point for a new event

THEN

(1) create a new E-MOP at that point
(2) extract the similarities of the two events and add those as content frame features of the new E-MOP
(3) index the two events in that E-MOP according to their differences from its norms

Figure 7-4

Does this reminding and generalization process mimic the human process? Imagine a person who has been to a French restaurant once in his life. According to the rules for indexing a first event, we would imagine it to be indexed under "serves French food" in his "restaurant" MOP. When he goes to a French restaurant a second time, we would expect him to remember the first time so that he can use that experience to predict the sequence of events of the second one. Thus, he will be reminded of the first experience in a French restaurant. If there are other similarities between the experiences, we would expect him to make generalizations based on those similarities. Thus, if both restaurants were fancy, and if in both there was good quiche as an appetizer, then he might make the generalizations that French restaurants are fancy and that they are a good place to have quiche. He would have to store that generalization somewhere, and the likely place to store it would be on a newly-created E-MOP "restaurant which serves French food", an E-MOP created from the two experiences.

When CYRUS indexes a second event where a first is already indexed, it retrieves the first episode, notices similarities between the two episodes and creates a new E-MOP. The second time Vance meets about military aid, for example, CYRUS is reminded of the prior meeting because both have the same topic. It checks the descriptions of both to see what other similarities they have. Suppose that both are with defense ministers. CYRUS concludes that "meetings about military aid are usually with defense ministers". It also creates a new "meetings

about military aid" MOP and indexes the two meetings within that E-MOP. Below is CYRUS' response to a second meeting about military aid, this one with the defense minister of Israel in Jerusalem.

```
Adding   $MEET actor (Vance)
               others (defense minister of Israel)
               topic (Military aid to Israel)
               place (Jerusalem)
to memory ...

Reminded of  $MEET actor (Vance)
                   others (defense minister of Pakistan)
                   topic (Military aid to Pakistan)
                   place (Washington)

because both are "diplomatic meetings"
        both have contract topic "military aid"

    creating new MOP:  meetings about military aid
  generalizing that when
    Vance meets about military aid,
    often he meets with a defense minister
```

Figure 7-5

After adding this event to memory, the "diplomatic meetings" MOP, which started out as in figure (7-3) above, will look like figure (7-6) below. This event, EV9, will be indexed as an event with underlying topic military aid, just as event EV7 with the Pakistani defense minister was indexed. That will cause reminding of EV7, and a new E-MOP "diplomatic meetings about military aid" will be created at that point (MOP12 below). Event EV9 will also be indexed in MOP10, "diplomatic meetings with defense ministers", and also in MOP6 as a meeting with an Israeli.

```
                 "diplomatic meetings" -- $MEET

content frame:  included in "negotiations"
                participants are foreign diplomats
                topic is an international contract
                is political and occupational for Vance
               /           |           |         \
diffs:        /            |           |          \
        underlying         |     participants      |
         topic  \          |     nationality       |
       /    |  \  \        |       /   | \         |
peace       |   \  trade   |     Israel | Arab     |
(MOP3)      |    \ (EV2)   |     (MOP6) | (MOP7)   |
    disarmament  \         |            |          |
     (MOP4)   mil aid      |        Pakistan       |
              (MOP12)      |          (EV7)        |
                      participants           included in
                      occupations                 |
                     /    |     \               summit
                    /  prim min  \            conference
               for min  (MOP9)    def min       (MOP8)
               (MOP5)             (MOP10)
```

Figure 7-6

The "diplomatic meetings" MOP, then, will look the same as before adding this meeting except that a new E-MOP will have been created at its index for underlying topic being "military aid". Other changes in organization will occur in the new E-MOP and in other E-MOPs this event was added to -- "diplomatic meetings with Israelis" (MOP6 above) and "diplomatic meetings with defense ministers" (MOP10 above).

What does the newly created E-MOP, "diplomatic meetings about military aid" (MOP12 above), look like? This new E-MOP, like "diplomatic meetings" has a content frame, and indexes its episodes according to their differences from those norms. Its content frame is a specialization of the content frame of "diplomatic meetings", containing only those aspects that are more specialized or different from that content frame. Thus, it will contain the information that these meetings are about military aid and are usually with defense ministers, two aspects its meetings (EV7 and EV9) have in common that are not specific to diplomatic meetings. The two events will be indexed in this E-MOP according to their differences from these norms. One of those is the nationalities of the participants. Another is the sides involved in the contract under discussion. Thus, the new E-MOP has the following content frame and indices:

```
        "diplomatic meetings about military aid" -- MOP12

  content frame: specialization of "diplomatic meeting"
                 participants are defense ministers
                 underlying topic is military aid
                 contract includes US as one side
  differences:       |                   \
                     |                    \
               participants´          sides of contract
               nationality               /        \
               /        \             Israel     Pakistan
            Israel     Pakistan        (EV9)        (EV7)
            (EV9)       (EV7)
```

Figure 7-7

Later, if CYRUS hears about a third meeting whose topic is military aid, it will assume that the meeting is with the defense minister of the country requesting aid (unless given contrary information). If asked for the participants of that third event, it will be able to answer "probably the defense minister".

7.3.3 E-MOP refinement

All event features do not fall into the categories described above. Often, aspects of an event´s description will be common to many past events. In that case, there will already be a sub-MOP at the point where that event is being indexed. In the example above, Vance´s meeting with the Israeli defense minister was directed to the E-MOPs "diplomatic meetings with defense ministers" and "diplomatic meetings with Israelis", both specializations of "diplomatic meetings".

In that case, two things happen during the processing. The new event gets indexed in the sub-MOP using the same procedures that indexed it in the more general E-MOP. In addition, the description of the new event is compared against the generalized information of both the parent E-MOP and the sub-MOP, and previous generalizations on both are confirmed or disconfirmed. This process is called E-MOP refinement.

E-MOP Refinement

IF there is an E-MOP at an index point for a new event

THEN

(1) index the event in that E-MOP
(2) check the validity of its generalizations
(3) update its generalizations as necessary

Figure 7-8

Imagine a person going a third time to a French restaurant. The third and subsequent times a person goes to a French restaurant, the discrimination "serves French food" will direct the episode to the "restaurants which serve French food" sub-MOP of the "going to a restaurant" MOP (the one we imagined being created in the section above). If the first two French restaurants a person went to were fancy, for example, then we would have expected him to make the generalization that "restaurants which serve French food are fancy". If the third one he went to were also fancy, it would serve to strengthen that generalization. If, however, it were not fancy, it would weaken the generalization. The person would no longer automatically expect a French restaurant to be fancy. If later visits to French restaurants proved that most French restaurants were fancy, the generalization would be strengthened again, and the non-fancy restaurant could be stored as an exception.

CYRUS, also, confirms and disconfirms generalizations as new events are added to its memory. On entering another meeting about military aid to memory, CYRUS will index it among other events already indexed there. Thus, a new meeting about military aid will be entered into the "meetings about military aid" sub-MOP of "diplomatic meetings" and will be indexed within that E-MOP (case 3).

In this way, reminding, generalization, and new E-MOP creation occur within newly created E-MOPs. If a new meeting about military aid to Pakistan is added to the E-MOP "meetings about military aid", illustrated in figure (7-7) above, CYRUS will be reminded of the first because both will be indexed as "Pakistan was one side of the contract discussed", and a new E-MOP, "diplomatic meetings about military aid to Pakistan" would be formed, a specialization of "diplomatic meetings about military aid".

Notice that the choice of an index and the choice of a sub-MOP to put an event into are the same. When a second event is indexed in the same way as a previous event, reminding happens and a new E-MOP is formed. In a sense, then, adding a new event to an E-MOP is like reminding, except that "reminding" is of a generalized group of events as opposed to a particular event. Another difference is that upon

"event reminding", a new E-MOP and new generalization are added to memory; upon "E-MOP reminding", those generalizations are checked. The process that go into confirming and disconfirming generalizations will be explained in a later section of this chapter.

7.4 Why generalize?

Indexing and sub-indexing could certainly be done without the use of generalized information. Why, then, should new E-MOPs be created and generalized knowledge be built up? Recall that in retrieving events from memory, elaboration is often necessary. Elaboration uses generalized information associated with E-MOPs. If that information were not added to newly created E-MOPs, elaboration could not be guided. Suppose, for example, that a "diplomatic meetings about military aid" MOP did not hold the information that their participants were usually defense ministers. How could the following question be answered?

(Q7-1) Who has Vance talked to about military aid?

In answering this question, the question "what countries have been in need of military aid recently?" might first be answered. Once those countries have been enumerated, however, it must be deduced which representatives of those countries would probably have been at meetings. Using the information that they usually include defense ministers, meeting participants can be found easily by recalling the defense ministers of each enumerated country and checking for meetings with each. Without the information that these meetings are usually with defense ministers, processing would have to proceed differently. Retrieval might have to include enumeration of the top officials of the enumerated countries, instead of simple defense minister look-up.

This question probably could be answered without the specific generalization about defense ministers. There are two important points to make, however. First, retrieval is easier using the information about defense ministers. Second, if the generalization about defense ministers was not made, other generalizations would also not have been made. Without generalized information, elaboration cannot be done. Since there are cases when elaboration is necessary for retrieval (see chapter 2), and elaboration cannot be done without generalized information, retrieval would have to fail in many cases.

There is a second important reason why generalized information on new E-MOPs is necessary. Recall that features in the content frame of an E-MOP are not indexed. By making generalizations and building up content frame information on new E-MOPs, later indexing in those E-MOPs is constrained. Suppose, again that the E-MOP "meetings about military aid" did not have the generalized information that its participants are usually defense ministers. Each time a new meeting about military aid were indexed at that point in memory, it would be further sub-indexed as a "meeting about military aid with a defense minister". If, in fact, all or most meetings about military aid were with defense ministers, then all of the events indexed as "meetings about military aid" would also be indexed as "meetings about military aid with defense ministers",

and the indexing under each of those points would be the same. Memory would have a lot of unneeded redundancy.

Third, generalized information can be used in generating default answers to questions. Suppose it was not known for a particular meeting who the meeting was with, but only the nationality of the participant was known. In other words, suppose the following meeting were entered into memory:

EV7-2: Vance met with a Jordanian about military aid.

This meeting would be indexed in the "meetings about military aid" MOP. Suppose the following question were later asked of the memory.

(Q7-2) Which Jordanian has Vance met with about military aid?

Retrieving EV7-2 would not answer this question since an actual person was being asked for. Using generalized information on the "meetings about military aid" MOP, however, the answer "probably the Jordanian defense minister" could be given.

7.5 Monitoring the usefulness of indices and norms

Because initial generalizations made in creating an E-MOP might be incomplete, inaccurate, or wrong, the update process must be able to monitor the correctness of generalized information in newly created E-MOPs. Checking the correctness of generalizations requires monitoring both the norms and indices in an E-MOP. When most of the events in an E-MOP have a particular index, the corresponding sub-MOP should be collapsed and its norms added to those of the parent MOP. When an E-MOP's norm fails to correspond to new events, that norm must be removed and an E-MOP built for it.

Index monitoring is also necessary as a way of keeping track of a feature's predictive power. If a particular feature does not correlate with other features, it should be marked as non-predictive so it will no longer be indexed. In this section, the processes of recovery from over-generalization or false generalization, generalization beyond the initial ones, and recovery from non-predictive indices will be discussed in detail.

7.5.1 Generalization -- when to do it

Recall that generalization is necessary for elaboration during retrieval and for control of later indexing and new E-MOP creation. One way generalization processes can be triggered is through reminding. When reminding happens, initial generalizations are made, and new E-MOPs are created. Generalization can also be triggered as a result of monitoring E-MOP indices and content frame features. When a particular event feature is true of nearly all events in an E-MOP, but is not yet a content frame feature, generalization procedures must be performed. There are four generalization-related processes that go on during memory update.

1. Similarities between two episodes are calculated and marked as E-MOP norms.

2. A content frame feature's certainty can be increased.

3. A content frame feature's certainty can be decreased.

4. An E-MOP index which indexes a large majority of the E-MOP's events can be collapsed, and its generalizations merged with those of its parent E-MOP.

7.5.1.1 Initial generalization

When event reminding occurs, there are two events available for comparison, creation of a new E-MOP, and generalization -- the current one and the previous one it is similar to. Each of those events has a set of features that describe it. Those features are compared to each other, and any that are in common are initially added to the content frame or norms of the new E-MOP.

Consider, for example, two trips that Vance might go on to the Middle East, one to Israel, and one to Egypt, both to negotiate Arab-Israeli peace. Suppose that in both he talked to the head of state of the country he was visiting, and in both, he was treated to a state dinner. Suppose both of those events were indexed as "trips to the Middle East" in the "diplomatic trips" MOP. What would the new "trips to the Middle East" MOP look like if these two trips were its initial two events?

Consider the generalized information we have available about "diplomatic trips". They are usually for the purpose of negotiations, include diplomatic meetings with diplomats of the foreign country, and are political and occupational. Consider, now, the specifications we have of the two trips:

```
----------------------------------------------------------------------

TRIP1:  destination:  Israel
        purpose:  negotiate Arab Israeli peace
        seq of events:  included meeting with Begin
                        included state dinner

----------------------------------------------------------------------
```

```
----------------------------------------------------------------------

TRIP2:  destination:  Egypt
        purpose:  negotiate Arab Israeli peace
        seq of events:  included meeting with Sadat
                        included state dinner

----------------------------------------------------------------------
```

The trips are similar in that both were to the Middle East, both had the same purpose, both included meetings with the head of state, and both included state dinners. Their differences were that each was to a different country, and each meeting was with a different head of state.

In creating a new E-MOP "diplomatic trips to the Middle East" from these two trips, the content frame of the new E-MOP should contain all of the similarities between the two events that are not true of diplomatic trips in general. Diplomatic trips generally include meetings with diplomats and are for the purpose of negotiating. However, because the negotiations and diplomatic meetings associated with these two trips are specializations of those "diplomatic trip" content frame properties, they will be good content frame norms for "diplomatic trips to the Middle East". Because their sequence of events both included state dinners, not true of all trips, that too, can be added to the content frame of the new E-MOP.

Thus, a new "diplomatic trips to the Middle East" MOP created from these two trips would have the following structure:

```
                "diplomatic trips to the Middle East"

      content frame:  destination is the Middle East
                      purpose is to negotiate Arab-Israeli peace
                      includes meeting with head of state
                      includes state dinner
                      specialization of "diplomatic trip"
                                   |
      differences:                 |
                              destination
                                /     \
                               /       \
                          Israel      Egypt
```

Figure 7-9

Some generalizations produced by extracting the similarities between these two events are more reasonable than others. All meetings indexed in this E-MOP will have "place = the Middle East" since that is the index for this sub-MOP in "diplomatic trip". Probably, these trips will continue to have the purpose of negotiating Arab-Israeli peace, at least as long as there is no peace there. We would not, however, expect that every trip to the Middle East will include a state dinner. As additional meetings are added to the E-MOP, the unreasonable generalizations will be disconfirmed and removed from the content frame. The processes for doing that will be explained in the remainder of this chapter.

7.5.1.2 Adjusting the certainty of a generalization

In some senses, the first two events added to an E-MOP are special. They are the events which are used to initially build an E-MOP's content frame. There are two possible problems with this method of generalization. First, some similarities between the first two events might be purely coincidental. Perhaps, in the example, above, it is only coincidental that both trips included state dinners. In that case, the program (and probably a person too) jumps to a wrong conclusion. The feature does not really belong in the content frame of the E-MOP. It is not a real generalization.

The second problem which could occur in using the first two events to build the content frame of an E-MOP is that one or both of the events might be lacking a feature that events of their type normally have. Perhaps later "trips to the Middle East" will always include a meeting with the head of the opposition party, a feature which the first two events did not have for some reason. In that case, the later generalization must be added to the E-MOP's content frame at a later time.

Because initial generalizations might be purely coincidental, there must be a way of confirming and disconfirming them. One way to do that is to check each new event as it is processed, to see if it conforms to the E-MOPs generalizations. If a feature of a new event conforms to a generalization, the certainty of that aspect of the E-MOP's content frame will increase. The certainty of each content frame feature that has a conflicting value in a new event will decrease. When the certainty of an aspect reaches a certain threshold, it can then be considered an actual norm for the E-MOP, or a real generalization. When a low threshold is reached, that aspect of the E-MOP's description need no longer be considered active for comparison.

We can think, then, of three classes of E-MOP features:

1. norms or actual generalizations

2. potential generalizations or active possibilities

3. E-MOP indices -- not potential or active possibilities for generalization

When an aspect is an active possibility or potential generalization (2), each new event added must be checked for confirming or disconfirming evidence of that aspect. Eventually, the aspect will become a norm (1) or a more general index (3), and not an active generalization possibility. At that point, it will no longer have to be monitored.

In CYRUS, the certainty of a generalization is a function of the number of events indexed in an E-MOP and the number of events with features which conflict with that generalization. Until the E-MOP organizes a reasonable number of events, however (6 in CYRUS), generalization certainty is not considered. This is an implementation detail which lends stability to an E-MOP until it can stabilize itself. As soon as an E-MOP reaches a reasonable size, the certainties of the generalizations are evaluated and those which fall below a threshold are removed, while all others remain.

All features that index an E-MOP in its parent MOP are considered norms (1) of the E-MOP immediately upon E-MOP creation. Since these features index the E-MOP, there will never be an event indexed in the E-MOP which does not have that feature. The E-MOP "diplomatic trips to the Middle East", for example, is indexed in the E-MOP "diplomatic trips" by its destination (the Middle East). Any event indexed in the E-MOP "diplomatic trips to the Middle East" will have to have that feature. Thus, it was a norm for the E-MOP "diplomatic trips to the Middle East" as soon as the E-MOP was created.

Other similarities between the first two events are considered potential content frame properties (2) until the E-MOP has grown to a reasonable size. At that time, norms that do not fit other E-MOP events are removed, and features that are true of a majority of events, but were not true of the first two are added to the content frame. Thus, if later trips to the Middle East do not include state dinners, that will

be removed from the E-MOP's content frame. If all trips to the Middle East following the two initial ones include a parade immediately after arrival, that would be added to the content frame. The next section will explain how that is done.

7.5.1.3 Collapsing sub-MOPs

The first two events indexed in an E-MOP are special -- they determine initially what the active generalization considerations are. Suppose, however, that the second and third events, or some later events, have common features that belong in the E-MOP's norms. How can those similarities be added to the content frame?

Suppose, for example, that in the first two museums a person went to, the exhibits were paintings. One was what we know of as a typical museum, where he walked around looking at the exhibits. The other was an avant-garde museum where he sat in a chair and the exhibits came by on a belt. What kind of generalizations could he make? If he had not been to other museums before, then we could not expect him to know that the second museum was weird. Thus, the only generalization we could expect him to make would be that "museums are places to see paintings". We would also expect him to recall his first experience as an experience "walking around to see the paintings" and his second as "exhibits were brought around to me". Suppose, further, that on later museum trips, he went to a number of typical museums, but with exhibits more varied than paintings. He would have to give up his generalization that "museums are places to see paintings" in favor of "museums are places to see art work". We would also expect him to update his generalizations about the sequence of events in a museum to include "walking around to see the exhibits", realizing at some point that his second museum experience with the conveyor belt was weird, and from then on recalling it that way.

How can the generalization that "normally in a museum one walks around to see the exhibits" be made? The reason this is problematic is that it was not a common aspect of the first two visits to museums. We would expect the next museum trip in which he went to a museum and walked around to remind him of his first museum trip. In CYRUS, that would trigger creation of a "going to a museum" sub-MOP specified as "going to a museum and walking around seeing exhibits". Subsequent museum trips would fall into that sub-MOP, and eventually it would grow to contain almost all museum trips. Because it would contain almost all trips, its indices would be practically the same as those for "going to a museum".

In order to automatically produce a more economical memory organization, indexing has to be monitored. Upon monitoring the indexing, it would be possible to notice that almost all museum experiences were falling into that sub-MOP and that it and its generalizations must therefore be the norms for museum trips. If that were done, the generalizations for "going to a museum and walking around" could be moved to the "going to a museum" MOP as its norms, and it and its sub-MOPs could be collapsed. Nothing would be lost since

each museum trip in the "going to a museum and walking around seeing exhibits" sub-MOP would have been indexed in other ways in the more general "going to a museum" MOP. A lot would be gained, because as more generalizations were added to the more general E-MOP, less indexing would have to be done. Thus, there would be a saving in both space and time.

Constant monitoring of an E-MOP's norms and indices will enable processing of this sort. Unless we want to consider all sub-MOPs of an E-MOP as potential generalizations, however, we need a rule for determining when sub-MOPs should be collapsed and their generalizations merged with those of their parent E-MOPs. The rule we might consider is the following:

Collapsing sub-MOPs

IF a sub-MOP indexes a large majority of the events in its parent E-MOP, and if the parent E-MOP is a reasonable size

THEN

(1) collapse the sub-MOP
(2) get rid of its index
(3) add the indexed feature plus other content frame features of the sub-MOP to the content frame of the parent E-MOP

Figure 7-10

This rule necessitates a way either of measuring the size of an E-MOP or of keeping track of relative sizes of E-MOPs and their sub-MOPs as updating is done. In CYRUS, the sizes of E-MOPs and their sub-MOPs is kept until the E-MOPs become large, at which time they are marked as "large".

CYRUS uses this rule to decide when to collapse specialized E-MOPs and move their generalizations to the parent E-MOP. After an E-MOP reaches a reasonable size, CYRUS checks each sub-MOP referred to by incoming events to see if any index a large majority of the events in the E-MOP. If one does, CYRUS collapses it and merges its generalizations with those of the parent E-MOP.

7.5.2 Recovery from bad generalizations

Every generalization made will not be good. An initial generalization, for example, could be coincidental. New information and events might contradict a previously made generalization. That means there must be a way of recovering from bad generalizations. In an example above, we illustrated how CYRUS would make the generalization that "trips to the Middle East include a state dinner". We also stated that after processing additional trips to the Middle East, CYRUS might discover that that was a bad generalization (i.e., it doesn't fit other episodes indexed in the E-MOP). In that case, the generalization would have to be removed.

Sometimes a generalization will have to be removed because a sub-MOP that has just been collapsed contains a generalization that contradicts a previous generalization on the parent E-MOP. Because the sub-MOP holds a large majority of the events in the parent E-MOP, its generalizations should be believed, and those on the parent should be doubted and removed if they are different.

This raises a special problem. Recall that while a feature is in the content frame of an E-MOP, events can never be indexed by that feature. Recall, also, that if a feature is in the content frame of an E-MOP, then there are at least some events which ought to have it as an index. Because events were not indexed by that feature, however, it would be impossible to go back and find all events supporting the generalization.

Generalization removal, then, can have grave implications in retrieval. Suppose a retrieval specification specified a feature that had been removed as a generalization, but which had not yet been indexed. If that happened, retrieval processes would not be able to find any trace in memory that an event with that feature had ever been processed. Using the criterion described in chapter 2 for determining that an event was not in memory, the retrieval processes would have to return a false answer.

Obviously, this shouldn't happen. If, each time a generalization were removed from the content frame of an E-MOP, an index was created for it in that E-MOP, then this would not have to happen. Retrieval processes would be able to return a message saying they could not find a specific event, but there have been some. Clearly, this would be preferable to returning nothing.

Old generalizations, then, must be indexed -- even if there are no distinctly known events that correspond to them. A sub-MOP indexed by the old generalization must be created with as much generalized knowledge as possible. Where can that generalized knowledge come from? If more than one generalization is being removed from an E-MOP at the same time, it is possible that the other generalizations being removed might be able to be associated with each other. When CYRUS, for example, recovers from its generalization that "diplomatic meetings are about Arab-Israeli peace", it is recovering at the same time from generalizations that diplomatic meetings involve Israelis and Egyptians. These two generalizations go together. When an E-MOP for "diplomatic

meetings about Arab-Israeli peace" is created, the other falsely generalized information can be associated with it as its generalized information.

Suppose, after recovering from the generalization that diplomatic meetings are about Arab-Israeli peace, memory is asked to retrieve "a meeting about Arab-Israeli peace with Begin in Jerusalem". While "diplomatic meetings" was marked as having topic Arab-Israeli peace, sub-MOPs of "diplomatic meetings" did not index that feature. Although there may be an E-MOP, then, for "diplomatic meetings with Begin", it will have no index for "topic is Arab-Israeli peace", nor will it have that feature marked as one of its norms (since it was a norm of a parent MOP). Thus, this target event will not be retrievable from the E-MOP "diplomatic meetings with Begin", or from any other sub-MOP of "diplomatic meetings" indexed while "topic is Arab-Israeli peace" was in its content frame.

The meeting with Begin will also not be retrievable from the newly-created "diplomatic meetings about Arab-Israeli peace" MOP, since that E-MOP will not yet have any individual events indexed in it. Thus, although there will be a "diplomatic meetings about Arab-Israeli peace" MOP, the retrieval strategies will find no evidence that any of those meetings were with Begin, or that any meetings with Begin were about Arab-Israeli peace. It will have to return with the message that there never was any such meeting.

One more procedure must happen, then, during recovery from bad generalizations. It would be nice, if after recovering from a bad generalization, all sub-MOPs of the E-MOP the generalization *was on* could be marked as possibly having that feature. Because E-MOP indices are not traversable without specification of an index, however, this would not be possible. The alternative solution to this problem is to mark the E-MOPs created for recovered generalizations as having been "once generalized". In that way, the retrieval functions will be able to come back with the message "there may be events with this description, but I can't find particular ones", instead of failing completely if no distinct event could be found.

The following rule summarizes the process of recovery from false generalization.

Recovery From False Generalization

IF a content frame feature has been disconfirmed

THEN

(1) remove it from the content frame
(2) create an empty E-MOP for it indexed by that feature
(3) add other features removed from the content frame at the same time
(4) mark the new E-MOP as "once generalized"

Figure 7-11

After a sub-MOP has been created from a recovered generalization, events that fit that sub-MOP can be indexed there. Those that had not been indexed while it was a generalization, however, would not be retrievable using that index.

We can now point out an important need for search strategies as part of the retrieval process. Although false generalization on one E-MOP might keep a particular event from being well-indexed, events it was related to might have been more richly indexed. If that is the case, they will be easier to retrieve than the event whose features had been falsely generalized. Finding a "once-generalized" E-MOP during traversal, then, should signal that search strategies will be particularly appropriate.

7.5.3 Maintenance of predictive power judgements

The last kind of monitoring that must be done during memory update is monitoring for predictive power of features. The fact that a particular type of feature is predictive is inherited from the E-MOP classifications and more general E-MOPs an E-MOP belongs to. Particular values of some type of feature, however, might turn out not to be predictive.

In CYRUS, the occupations of event participants is marked as a predictive feature for all occupational events. Some occupations, however, are better than others at predicting additional event features. "Defense ministers" in "diplomatic meetings", for example, normally meet about military aid. That occupation is therefore predictive for diplomatic meetings. "Foreign ministers" and "heads of state", however, are not predictive of any other meeting features.

In CYRUS, a feature is checked for predictive power after it has occurred a reasonable number of times (6 in CYRUS) to see if it has co-occurred with any other features. If the content frame of the E-MOP that a feature indexes contains only those features which index the E-MOP and no others, then the feature is judged not to be predictive, and is marked as such. After being marked as non-predictive, a feature is no longer indexed.

Why shouldn´t non-predictive features be indexed? Recall that one of the reasons for building a content frame on newly created E-MOPs is to keep track of the similarities between the events and to constrain later indexing. An E-MOP indexed by a feature which is non-predictive will have no generalized information to keep track of, and it will not be able to constrain indexing and new E-MOP creation. Thus, its indices will duplicate the indices in its parent E-MOP without it keeping track of useful generalized information. It will waste space without contributing to the organization of generalized information.

7.6 How CYRUS´ memory grows

Initially, CYRUS´ "diplomatic meetings" MOP has no indices, but does have a content frame with generalized information about diplomatic meetings. It knows that diplomatic meetings are generally instrumental to negotiations, that their participants are foreign diplomats, that their topic of discussion is usually some international contract or dispute, and that they are political and occupational activities for Vance. Political activities for Vance are activities involving diplomats, whose topics of discussion are usually international relations of some sort. "Diplomatic meetings", then, begins the following content frame:

"diplomatic meetings" -- $MEET

```
content frame:  included in "negotiations"
                participants are foreign diplomats
                topic is an international contract
                topic involves the United States
                is political and occupational for Vance
                is a meeting
```

Figure 7-12

After adding two "diplomatic meetings" to CYRUS, one concerning SALT with Gromyko (EV2), and one concerning Arab-Israeli peace with Begin (EV1), both part of negotiations, CYRUS´ "diplomatic meetings" MOP includes the following structure:

```
                 "diplomatic meetings" -- $MEET

  content frame:  included in "negotiations"
                  participants are foreign diplomats
                  topic is an international contract
                  topic involves the United States
                  is political and occupational for Vance
                /     |        |          |     \
  diffs:       /      |        |          |      \
              /       |    participants   |     underlying
          topic       |     nationality   |       topic
         /  \         |       /    \      |      /    \
        /   Salt      |   Israel   USSR   |    peace   \
    Arab-   |         |      |     |      |      |    arms
    Israel  EV2    topic's  EV1   EV2     |     EV1  limits
    peace          sides                  |             |
     |             /   \             participants'     EV2
    EV1           /    USSR          occupations
               Israel    |            /      \
               & Arabs  EV2         foreign   \
                 |                  minister  head
                EV1                   |       of state
                                     EV2        |
                                               EV1
```

Figure 7-13

These two meetings are indexed by their topics, the issue underlying those topics, the sides involved, and the the nationalities and occupations of the participants. Because both are part of negotiations standard to the topic they are discussing, neither are indexed by their purpose or larger episodes they are part of. Because these two meetings have nothing in common except generalized information already known about meetings, they are each indexed uniquely by their differences, and "diplomatic meeting" content frame information does not change.

Thus, all indices in this E-MOP index event descriptions. In CYRUS, and in the E-MOP pictured above, those event descriptions are pointers to events themselves. In a more reconstructive memory, each event description could be the set of features of the meeting which were different than the norms. Other unspecified features could be inferred through E-MOP content frame information. The event description indexed under "meeting with a head of state" would include the information that the meeting was about Arab-Israeli peace, and that it was with Begin, an Israeli. The event description indexed under "meeting about Arab-Israeli peace" would include the information that the meeting was with Begin, the Israeli prime minister.

Both of these meetings were meetings that happened in August, 1978. If we continue adding meetings to CYRUS in the order in which they happened, we can illustrate how the structure of the "diplomatic meeting" MOP changes over time. The next meeting processed by CYRUS was another meeting about Arab-Israeli peace, this time with Sadat. Because this meeting had features in common with Vance's meeting with Begin, reminding occurred, new E-MOPs were created, and generalized information was added to those E-MOPs. The content frame of "diplomatic meetings" remained the same, since there were no additional overwhelming similarities between the three meetings indexed up to that point. After adding this meeting to memory, CYRUS' "diplomatic meeting" MOP had the following structure:

```
               "diplomatic meetings" -- $MEET

content frame:  included in "negotiations"
                participants are foreign diplomats
                topic is an international contract
                topic involves the United States
                is political and occupational for Vance
              /      |          |         |     \
diffs:       /       |          |         |      \
            /        |    participants    |    underlying
        topic        |    nationality     |      topic
        /  \         |      /    |  \     |      /    \
       /   Salt      |  Israel   | USSR   |   peace    \
  Arab-     |        |      |    |    |   |  (MOP4)   arms
  Israel   EV2   topic's   EV1   |  EV2   |          limits
  peace           sides        Egypt      |              |
  (MOP1)         /   \            |       |            EV2
                /   USSR        EV3       |
            Israel     |            participants'
            & Arabs  EV2            occupations
            (MOP2)                    /      \
                                   foreign    \
                                   minister  head
                                      |      of state
                                    EV2         |
                                              EV1
```

Figure 7-14

Because this meeting, too, was with a head of state, and was about Arab-Israeli peace, new E-MOPs were created at those index points (MOP1, MOP2, MOP3, and MOP4 above). Those E-MOPs each contain as norms the information common to the two meetings which initiated their creation -- in this case, that both were about Arab-Israeli peace, and both were with a head of state. Each one also is marked as a specialization of "diplomatic meeting". The newly-created E-MOP "diplomatic meetings about Arab-Israeli peace (MOP1), for example, has the following

structure:

```
   "diplomatic meetings about Arab-Israeli peace" -- MOP1

content frame:  topic is Arab-Israeli peace
                underlying topic is peace
                involves Israel and the Arabs
                participants are heads of state
                          |
differences:              |
                    participants'
                    nationality
                     /       \
                    /         \
                 Israel     Egypt
```

Figure 7-15

When CYRUS finds an index which indexes nearly all the events in the E-MOP, it predicts that that feature should actually be one of the E-MOP's norms, and updates the content frame appropriately. As additional diplomatic meetings were added to CYRUS' memory, its "diplomatic meetings" MOP continued to grow. The next 8 meetings Vance had were meetings concerning Arab-Israeli peace. Some were part of the Camp David Summit. Because so many of the diplomatic meetings CYRUS knew about were about Arab-Israeli peace, and because there was only one meeting that had been about anything else at that point, CYRUS made the generalization that "Vance's diplomatic meetings are generally about Arab-Israeli peace". Thus, after adding those eight meetings to memory, CYRUS' "diplomatic meetings" MOP had the following content frame and structure:

```
               "diplomatic meetings" -- $MEET

content frame: included in "negotiations"
               participants are foreign diplomats
               topic is an international contract
               topic involves the United States
               is political and occupational for Vance
               topic is Arab-Israeli peace
               underlying topic is peace
               involves Israel and the Arabs
           /   |          |           |    |      \
diffs:    /    |          |           |    |       \
         /     |     participants     |     \       \
      topic    |     nationality      |   included   \
    / \        |       /  |   | \     |     in        \
    /   Salt   | Israel   |   | USSR  |      |         |
Camp     |     | (MOP10)  |   |   |   |    summit      |
David   EV2    |       Egypc  |  EV2  |    (MOP9)      |
Accords        |      (MOP6)  |       |           underlying
(MOP5)         |           Jordan     |              topic
               |             |        |                |
               |            EV6       |                |
             topic's                  |               arms
              sides             participants'        limits
             /   \                occupations          |
            /    USSR          /     |   \            EV2
         Egypt     |       foreign   |    def min
         & Israel EV2      minister  |      |
         (MOP8)            (MOP7)   head   EV5
                                  of state
                                   (MOP3)
```

Figure 7-16

Because CYRUS had made the generalization that Vance's diplomatic meetings were about Arab-Israeli peace, it removed some of the sub-MOPs it had created previously. Thus, its E-MOPs "diplomatic meetings about Arab-Israeli peace", "diplomatic meetings about a topic involving the Arabs and Israel", and "diplomatic meetings about peace" (MOPs 1, 2, and 4 in Figure (7-14)) were collapsed. They were subsumed by the generalizations made about "diplomatic meetings" in general. Notice, however, that at this point in memory's growth, there were specialized E-MOPs with features specialized from those content frame features New E-MOPs for "diplomatic meetings about a topic involving Egypt and Israel" (MOP8) and "diplomatic meetings about the Camp David Accords" (MOP5) specialize the content frame property "topic involves Arab-Israeli peace". In addition, the E-MOPs "meetings with Israelis" (MOP10) and "meetings with Egyptians" (MOP6) were created. Removal of sub-MOPs from an E-MOP will be discussed in the next section.

At this point, CYRUS had made a number of erroneous generalizations about diplomatic meetings. It generalized that diplomatic meetings usually involve Arab and Israeli concerns, and are normally about peace between the Arabs and Israel. We recognize that the meetings CYRUS had seen up to this point were not a good sampling of his meetings. As far as CYRUS knew, however, the meetings it had seen were typical of Vance's meetings. To it, then, these were valid generalizations, and CYRUS used them to constrain further index and new E-MOP creation, and to guide elaboration during retrieval.

Suppose, at this point, we were to ask CYRUS the following:

(Q7-3) What have you talked about at diplomatic meetings?

In answering this question, CYRUS would first check topic specifications in the content frame of "diplomatic meetings". It would find that they are generally about Arab-Israeli peace. At that point, it would enumerate the Arab-Israeli topics it knew about -- the Camp David Accords. It would thus attempt retrieval of "diplomatic meetings about the Camp David Accords". It would also infer that participants were from the Middle East and search the E-MOP for meetings with people from each of the Middle Eastern countries. It would thus do a good job of elaborating on and retrieving meetings about Arab-Israeli peace, but would forget the SALT meeting entirely.

Suppose, however, that it were asked the following question:

(Q7-4) Who have you talked to about SALT?

Although elaboration at this point would not allow retrieval of a meeting about SALT, specification of its unique characteristic does allow its retrieval. Because an indexed feature unique to to one event in the "diplomatic meetings" MOP is specified in the question, CYRUS can retrieve the appropriate meeting by traversing the E-MOP, without the need to apply any search strategies.

In fact, this is an important observation about retrieval of events which are very different from an E-MOP's norms. They will not be retrievable through elaboration since norms will not aid in specifying their features. They will, however, be easily retrievable if their significant unique features are specified.

CYRUS' memory at this point was, at best, confused. It is not always possible to know when there is a good sampling. As a result, indices and norms must constantly be monitored to keep memory's generalized information up to date with the events it organizes.

When a content frame feature is no longer true of most events indexed in an E-MOP, CYRUS updates both its content frame and indices, removing that feature from its content frame and adding indices for it in the E-MOP. Those indices will not be able to point to actual events, but they will be point to the fact that there are a large number of events that have that feature. In that way, it recovers from false generalizations.

As additional meetings were added to CYRUS´ memory, it began to notice that very few of them matched the content frame that it had generalized. After approximately 20 meetings had been added to CYRUS´ "diplomatic meetings" MOP, its content frame became more stable. At that point, it had a good sampling, and the meetings balanced each other in all relevant ways. It recovered and readjusted its generalizations to fit reality. It removed the bad generalizations from the content frame, created indices for them, and used those features to index later meetings.

After CYRUS has processed approximately 60 diplomatic meetings, its "diplomatic meetings" MOP contained the following content frame, and included the following indices:

```
                     "diplomatic meetings" -- $MEET

    content frame:  included in "negotiations"
                    participants are foreign diplomats
                    topic is an international contract
                    topic involves the United States
                    is political and occupational for Vance
                 /  |          |          |   |     \
    diffs:      /   |          |          |   |      \
               /    |     participants    |    \      \
          topic     |     nationality     |  included  |
         /  |  \    |      / |   | \      |     in     |
        /   | Salt  | Israel |   | USSR   |     |      |
    Camp    |   |   | (MOP10)|   |  |     |   summit   |
    David   |  EV2  |     Egypt  | EV2    |   (MOP9)   |
    Acc     |       |    (MOP6)  |        |        underlying
    (MOP5)  |       |        Jordan       |          topic
        Arab-       |           |         |          /  |
        Israeli     |          EV6        |      peace  |
        peace   topic's                   |            arms
           .... sides               participants´      limits
            /   /   \               occupations         |
        Arabs  /   USSR                /    |   \       EV2
         &   Egypt   |           foreign    |   def min
      Israel & Israel EV2        minister   |     |
            (MOP8)               (MOP7)  head    EV5
                                       of state
                                        (MOP3)
```

Figure 7-17

The entire structure of CYRUS´ "diplomatic meetings" MOP after 60 meetings is not illustrated here. The interesting features to note, however, are that its content frame had recovered from the erroneous generalizations that were made, and it again had indices and corresponding E-MOPs for the features "topic involving Arabs and

Israel", "topic is peace", and "topic is Arab-Israeli peace". Those were the E-MOPs that had been removed when CYRUS made the previous generalizations.

When CYRUS finds an index with no predictive power, it marks that index as useless and no longer uses it. When CYRUS began adding new meetings to memory, it indexed all events according to the occupations of their participants. By the time it had added 60 meetings to memory, however, it knew that the occupation "foreign minister" was not predictive of other features, and it therefore no longer used them for indices in its "diplomatic meetings" MOP. Some of the sub-MOPs to "diplomatic meetings" also have features marked as non-predictive or useless.

As CYRUS is adding new events to memory, it monitors both its indices and its content frame, keeping its content frame up to date, and attempting to ascertain which indices are good and which are not useful. As a result of this monitoring, recovery from over-generalization or false generalization, generalizations beyond the initial ones, and recovery from non-predictive indices are possible.

7.7 Negative implications of memory reorganization

References from events to episodes they are related to are derived at the time the event is entered into memory. Those references are unique descriptions of the related event at the time the events are entered into memory. Because memory's organization changes over time, however, a reference that was unique at input time may not be unique at retrieval time. During retrieval, if the available specification of a contextually-related event is no longer unique, then elaboration must be applied to reconstruct additional details of the related event and enable its retrieval.

Consider, for example, the following event:

EV7-3: On his recent visit to the Philadelphia, Vance was welcomed by a marching band playing "Stars and Stripes Forever".

Suppose this particular welcoming ceremony was the first that had included a band. In that case, the trip would refer to the welcoming ceremony as "welcomed by a band". Since the welcoming ceremony itself would be stored in its own E-MOP, that specification would be a unique reference to the welcoming ceremony. Suppose, however, that during later trips around the United States, Vance was again greeted by bands, sometimes high school marching bands, sometimes college bands, sometimes military bands. The specification "welcomed by a band" associated with his visit to Philadelphia, would no longer be unique. It would describe an E-MOP rather than an individual event. To retrieve the particular welcoming ceremony, the specification "welcomed by a band" would have to be elaborated until it described a unique event.

Elaboration during retrieval, however, can be faulty. In that case, the wrong event might be retrieved from memory. If a recalled event is consistant with the information that is known, its differences will not be noticed, and false reconstruction will occur. Faulty retrieval of this kind corresponds to what psychologists refer to as _recall intrusion_ (Crowder, 1975, Owens, et al., 1979) and _recognition confusion_ (Bower, et al, 1979, Gibbs and Tenny, 1980) in people. Mistakes of this sort happen, in the retrieval process we have described, when some aspect of the reconstruction directs retrieval to the wrong episode, which is substituted for what actually happened.

Retrieval interferences occur because the constantly changing memory structure requires elaboration of specifications for retrieval of individual events during reconstruction. The features specified for related events depend on the memory organization at the time the event is entered into memory. They will be features which discriminate the related event in its E-MOP at that time. If memory changes before retrieval time, then what was once a unique specification of an event may no longer be unique.

7.8 Summary

When new events are added to memory, they must be added in such a way that memory's organization is maintained. Adding a new event to an E-MOP is a process of integrating it into memory. The event is indexed in the E-MOP so that it can later be retrieved, and generalized information necessary to aid its retrieval is built up.

The first step in adding a new event to an E-MOP is to choose appropriate features of the event for indexing. Each feature chosen can have one of three relationships to the E-MOP:

1. There is nothing yet indexed in the E-MOP with that feature

2. There is one other item with that feature indexed in the E-MOP

3. There is an E-MOP indexed by that feature

When there is not already an index for a feature (1), a new index is built, and the event is indexed at that point. When there is one other event with a particular feature (2), a new E-MOP is formed based on the similarities between the new event and the previous one with that feature, and the two events are indexed in that E-MOP. When there is already an E-MOP indexed by a particular feature (3), the new event is integrated into that E-MOP. That integration includes refining the E-MOP's generalized information and indexing the event in the E-MOP. Refining an E-MOP's generalizations involves both collapsing sub-MOPs and adding their generalized knowledge to that of the parent E-MOP, and removing a false generalization from the content frame of an E-MOP and creating an E-MOP for it.

The memory organization that is built up in adding new events to memory hierarchical organization of categories and their sub-categories. As new events are added to memory, they are multiply indexed in each of the categories or sub-categories they fit into. In many cases, this results in the creation of specializations of the sub-categories. The event being added to memory must then be indexed in those sub-sub-categories. In system which creates multiple new indices for each new event as it is added to memory, and which creates new sub-categories automatically, the number of indices and categories in memory would quickly reach astronomical proportions if it were not controlled.

Two ways of controlling creation of new indices and sub-MOPs have been presented. Because only differences from norms with predictive power are indexed, maintaining both generalized knowledge and knowledge about the potential predictive power of indices is necessary to prevent combinatorial explosion of indices and sub-categories. In this chapter, maintenance of both of those kinds of information has been explained.

Generalization maintenance is important to both the retrieval and updating processes. During retrieval, generalized information is needed to apply retrieval strategies. During memory update, generalized information constrains later indexing and new E-MOP creation.

AD-A093 084 YALE UNIV NEW HAVEN CT DEPT OF COMPUTER SCIENCE F/G 9/2
RETRIEVAL AND ORGANIZATIONAL STRATEGIES IN CONCEPTUAL MEMORY: A--ETC(U)
NOV 80 J L KOLODNER N00014-75-C-1111
UNCLASSIFIED RR-187 NL

3 of 3
AD A
093084

END
DATE
FILMED
2-81
DTIC

CHAPTER 8

How CYRUS works

8.1 Introduction

CYRUS currently has two data bases. The more detailed data base contains information about former U. S. Secretary of State Cyrus Vance. He was chosen because he was in the news often enough to generate a large number of news updates, but he did few enough types of events to make the initial specification of E-MOPs feasible. More recently, CYRUS has begun organizing information about U. S. Secretary of State Edmund Muskie into a second data base of information about Muskie. The initial memory organization CYRUS started with for each man was the same. However, because of their differing experiences, the new categories built by the system for Muskie's events are somewhat different than those built for Vance.

In one mode of operation, CYRUS is hooked up with FRUMP (DeJong, 1979) to form a complete information retrieval system called CyFr. FRUMP reads stories from the UPI news wire, and sends conceptual summaries of stories about Muskie and Vance to CYRUS. CYRUS then adds the new events to its memory and answers questions about them. CYRUS' Muskie memory has been built up entirely from FRUMP-processed stories. Its Vance memory is built partially of FRUMP-processed stories and partially of stories encoded by hand.

8.2 Retrieval in CYRUS

CYRUS retrieves answers to questions posed to it in English. Although the retrieval process described in previous chapters is the primary part of CYRUS' question-answering procedure, it is only one part of it. CYRUS' question-answering loop is based on Lehnert's (1978) description of the question answering process. In order to answer a question, it must be parsed and its conceptual category and question concept must be extracted. The *question concept* (Lehnert, 1978) of a question is the part of the question that must be searched for, i.e., its target event. The target event is searched for using the retrieval process already described, and when it is found, processes corresponding to the question category are applied to it to formulate an answer. If the question asks "why", for example, the causes of the retrieved event are extracted. An English language generator generates an answer. The entire question-answering process can be described by the following algorithm:

Question Answering in CYRUS

```
      +-----------------+
      | get a question  |  <------+
      +-----------------+         |
            \ /                   |
        +-----------+             |
        | parse it  |             |
        +-----------+             |
            \ /                   |
    +---------------------+       |
    | resolve references  |       |
    +---------------------+       |
            \ /                   |
  +------------------------+      |
  | extract question type  |      |
  |  and target concept    |      |
  +------------------------+      |
            \ /                   |
      +----------------+          |
      | search memory  |          |
      +----------------+          |
            \ /                   |
     +------------------+         |
     | formulate answer |         |
     +------------------+         |
            \ /                   |
     +-----------------+          |
     | generate answer | ---------+
     +-----------------+
```

Figure 8-1

In this section, parts of the question-answering cycle will be explained and illustrated.

8.2.1 Searching CYRUS' memory

CYRUS' memory is searched using the retrieval process described in chapter 2. Given a target concept, a context for search is constructed, the appropriate E-MOPs are traversed, and the target event is elaborated as necessary. If an appropriate answer has not been found, search strategies are applied to search for alternate related events. Because strategy application has been described in detail in previous chapters, the application of strategies themselves will not be described in this chapter. Rather, the subset of strategies implemented in CYRUS will be pointed out.

In previous chapters of the thesis, it was explained that E-MOPs keep track of their causal, containment and temporal relations to other E-MOPs. CYRUS' E-MOPs implement the containment and temporal relations, but do not keep track of causal information. Thus, CYRUS' E-MOPs specify larger episodes they usually occur in, and their normal sequence of events, standard methods of achievement, and temporal relationships within the event sequence of another E-MOP. An implication of this is that only those strategies associated with containment relationships are implemented and applied within CYRUS. Those strategies include "Find-from-Seq-of-Events", "Find-in-Simple-MOPs", "Find-in-IMOPs", and "Find-from-Standardizations".

Initial context construction involves using contextual information associated with components of the target concept to infer an E-MOP for traversal. CYRUS' context instantiation strategies include those associated with topics, participants, and places.

The traversal/elaboration procedure and most of the retrieval strategies that have been presented are implemented in CYRUS. When an E-MOP is reached during traversal, CYRUS applies component-instantiation rules to elaborate on the target concept and then continues its traversal. The shortest path to the target concept in memory retrieves it. Calls to component-instantiation rules are embedded in CYRUS' traversal procedure. When elaboration is necessary, choice of component-instantiation rules is driven by the types of indices the E-MOP to be traversed has. If it has "participant" indices, for example, the procedure PARTICIPANTS is applied to infer participants. Thus, CYRUS' component-instantiation rules are driven by the memory's organization.

When CYRUS applies search strategies, it applies those associated with the type of E-MOP being searched for. Thus, for scripts, it applies "Find-from-IMOPs" and "Find-from-Simple-MOPs". It continues applying search strategies until sufficient matches to the target event are found. A satisfactory answer has been found when CYRUS is asked for one example and finds one (as in verifying if something has happened), when it is asked for a particular number of episodes and finds that many, or after it has applied all the strategies it knows.

Examples of CYRUS' strategy application can be found in chapters 2, 3, and 4, and won't be repeated here.

8.2.2 Parsing questions

CYRUS' question parser is based on CA (Birnbaum and Selfridge, 1979). Martin Korsin extended that parser for CYRUS and added necessary word definitions. The parser takes an English question as input and produces a conceptual dependency representation of the meaning of the question. The parser has definitions for approximately 300 words, and can parse one and two clause questions. The following is an example of parser output:

Parser Output

```
>Has Vance talked to Gromyko about SALT recently?

parsing ...
((ACTOR HUM1 <=> (*MTRANS*)
             MOBJECT (*CONCEPTS* CONCERNING CNTRCT2)
             TO HUM66)
 TIME (#TIME TENSE (PAST))
 MODE (*?*))
```

Figure 8-2

After parsing a question, CYRUS searches memory to retrieve an answer as explained in previous chapters. It first applies context-instantiation strategies to infer a context for search. Because Gromyko is a foreign dignitary, CYRUS chooses $MEET ("diplomatic meetings") as its context for search in answering the question above, and searches for a recent meeting between Vance and Gromyko. It finds one, gives an answer, asks for the next question, and parses it:

```
The question is:
((ACTOR HUM1 <=> (*MTRANS*)
            MOBJECT (*CONCEPTS* CONCERNING CNTRCT2)
            TO HUM66)
 TIME G0852 MODE (*?*))
The question type is "verification"
The question concept is:
((ACTOR HUM1 <=> (*MTRANS*)
            MOBJECT (*CONCEPTS* CONCERNING CNTRCT2)
            TO HUM66) TIME G0852)

   inferring a diplomatic meeting

The answer is:
((<=> ($MEET ACTOR HUM1 OTHERS HUM66 TOPIC CNTRCT2))
 TIME G1058 PLACE POL21 DURATION G1061)
 Modality YES

 yes, most recently in Geneva in December.


>Did he talk to him about SALT?

parsing ...
((ACTOR (*PP* PPCLASS (#PERSON) GENDER (*MASC*) REF (DEF))
    <=> (*MTRANS*)
    MOBJECT (*CONCEPTS* CONCERNING CNTRCT2)
    TO (*PP* PPCLASS (#PERSON) GENDER (*MASC*) REF (DEF)))
 MODE (*?*) TIME (#TIME TENSE (PAST)))
```

When people ask questions, they often reference previous aspects of the context using pronouns. CYRUS' parser does not resolve those references, but marks pronominal references for the reference resolver to resolve. The question "Did he talk to him about SALT?" has two references which must be resolved.

8.2.3 Reference resolution in CYRUS

CYRUS' reference resolution processes combine pattern matching with a process of matching individual pronouns to references made in the previous question and answer. Any time a pronominal reference in a question can refer to only one component of the conceptual representation underlying the previous answer, CYRUS resolves the reference to that component.

Unfortunately, references do not always uniquely specify a component of a previous answer, as exemplified in the dialog above:

Q1: Has Vance met with Gromyko recently?
A1: Yes, most recently in Geneva in December.
Q2: Did he talk to him about SALT?

Because the answer to A1 implicitly specifies two men (Vance and Gromyko), the "he" and the "him" in the second question cannot be resolved by choosing the male person in the answer that came before it. The way CYRUS resolves the reference is to match the representation of the question with the representation of the answer that came before it. The following are the representations for A1 and Q2, repeated from above:

Reference resolution

```
A1:   ((<=> ($MEET ACTOR HUM1 OTHERS HUM66 TOPIC CNTRCT2))
        TIME G1058
        PLACE POL21
        DURATION G1061)

Q2:
   ((ACTOR (*PP* PPCLASS (#PERSON) GENDER (*MASC*) REF
(DEF))
        <=> (*MTRANS*)
        MOBJECT (*CONCEPTS* CONCERNING CNTRCT2)
        TO (*PP* PPCLASS (#PERSON) GENDER (*MASC*) REF
(DEF)))
     MODE (*?*) TIME (#TIME TENSE (PAST)))
```

Figure 8-3

There is an additional problem in resolving these references. The pronominal references in the question must be resolved by pattern matching, but the two patterns do not match each other. A1 specifies a "diplomatic meeting", while Q2 refers to an MTRANS (one of Schank's (1975) Conceptual Dependency primitives). Talking, however, can occur in many different contexts, including the context of a "diplomatic meeting" ($MEET). In order to resolve the reference in this case, the MAINCON of $MEET is instantiated. It is also an MTRANS, and the two MTRANSs are matched against each other. Following is an instantiation of the maincon of A1:

```
((ACTOR HUM1 <=> (*MTRANS*)
             MOBJECT (*CONCEPTS* CONCERNING CNTRCT2)
             TO HUM66)
 TIME G1058
 PLACE POL21)
```

Matching this to Q2, CYRUS resolves the references in Q2. The actor is understood to be HUM1 (Vance) and the TO slot is filled by HUM66

(Gromyko). Q2 can be understood as "Did Vance talk to Gromyko about SALT?" CYRUS' resolution of the references is illustrated below:

Reference resolved

```
resolving references ...
The question is:
((ACTOR HUM1 <=> (*MTRANS*)
             MOBJECT (*CONCEPTS* CONCERNING CNTRCT2)
             TO HUM66)
 MODE (*?*) TIME G0906)
```

Figure 8-4

The emphasis in designing CYRUS was on its memory organization and search. As a result, its other components are not as well developed. CYRUS' reference resolution procedure works for the sets of questions that have been given to CYRUS, but is not a general solution to reference handling.

8.2.4 *To search or not to search?*

In previous chapters, CYRUS' strategies for extensive search of memory have been described. Memory search, however, does not always need to be extensive. When a question references the answer to a previous question, it can be answered by searching the episodic context (see chapter *4*) of the question instead of searching all of memory. In the dialog above, A1 should be used to answer Q2 without searching memory for another answer.

The reference resolution process in CYRUS is instrumental in deciding whether the episodic context of a previous answer should be searched or whether memory should be searched extensively. If a question's references are all to components of the last answer, then the episodic context of that event should be searched to answer the question. Thus, in the example above, A1 is chosen for search because all of the references made in Q2 were to components of A1.

```
answering question using previous context:
 directly

The answer is:
((<=> ($MEET ACTOR HUM1 OTHERS HUM66 TOPIC CNTRCT2))
 TIME G0943 PLACE POL21 DURATION G0946)
 Modality YES

 yes, for 5 hours.
```

Reference resolution processes are also used to aid in choosing a context for search in CYRUS, as illustrated by this last question. Although Q2 does not specify a context for search, context instantiation strategies did not have to be applied to infer an E-MOP. Rather, a context for search can be inferred directly from the previous answer during the reference resolution process because its MAINCON matches the question concept. Thus, CYRUS infers that the "talking" referred to in the question is a reference to a "diplomatic meeting" without applying context instantiation strategies.

8.2.5 Search of episodic context

Once reference is used to decide that a question should be answered using the previous context, that episode must be searched for the target event. CYRUS uses the following algorithm to search that context:

Episodic Context Search in CYRUS

```
(1) does the target event match the
        previous answer?
      /                 \
    yes                  no
    /                     \
use that     (2) is the target event part of sequence
to answer        of events of the previous answer?
                    /        \
                  yes         no
                  /            \
          use that  (3)  does the target event correspond
          to answer      to one of the episodes the previous
                         context was part of?
                           /             \
                         yes              no
                         /                 \
                 use that    (4) does the target event correspond
                 to answer       to an event in the sequence of
                                 events of the episodes the previous
                                 answer was part of?
                                 /            \
                               yes             no
                               /                \
                       use that              search memory
                       to answer
```

Figure 8-5

In answering Q2, the first test in this procedure is satisfied, and the question is answered using the meeting context from the previous question. Consider, for example, the following continuation of the dialog:

Q2: Did he talk to him about SALT?
A2: Yes, for 5 hours.
Q3: When did he leave Geneva?

In answering this question, the first test fails. The second one is applied. Because $MEET (the previous answer's E-MOP) is a script, however, its sequence of events is not appropriate to search, and CYRUS applies the third test to check whether Vance's leaving Geneva is a larger event the meeting was part of. That test also fails, and it applies the last test, checking to see if "leaving Geneva" is part of the sequence of events that the meeting from A2 was part of. This test is successful, and CYRUS answers the question.

```
>When did he leave Geneva?

parsing ...
((ACTOR (*PP* PPCLASS (#PERSON) GENDER (*MASC*) REF (DEF))
        <=> (*PTRANS*)
        FROM POL21)
  TIME (*?*))
resolving references ...
The question is:
((ACTOR HUM1 <=> (*PTRANS*) FROM POL21) TIME (*?*))

The question type is "time"
The question concept is:
((ACTOR HUM1 <=> (*PTRANS*) FROM POL21))

answering question using previous context
 directly
 using top-level events

The answer is:
((<=> ($FLY ACTOR HUM1 ORIGIN POL21 DESTINATION POL2))
 TIME G1021)

  on December 24.
```

8.2.6 *Using time as a context for retrieval*

CYRUS also uses the above process to answer questions with embedded time specifications. It first searches for the embedded time specification, treats the event it finds as the context for retrieval, and applies the algorithm above to it to find the target. Consider, for example, the following question:

```
>Last time Vance was in Saudi Arabia, did he go sightseeing?

parsing ...
((<=> ($M-SIGHTSEE ACTOR HUM1))
  MODE (*?*)
  TIME (#TIME EVENT ((ACTOR HUM1 IS (*LOC* VAL POL11))
                     TIME (#TIME TENSE (PAST)))
                               NUMBER (-1)
                               TENSE (PAST)))
```

This question is composed of a time specification, "last time you were in Saudi Arabia", and a target event -- sightseeing. The representation the parser produces for this question embeds the time specification in the time slot of the target sightseeing event. The event representation in the time slot is the representation for "Vance was in Saudi Arabia" (HUM1 is Vance, POL11 is Saudi Arabia). The NUMBER slot of the TIME representation specifies that it is the last time that event happened that is of interest.

CYRUS answers this question by first finding Vance's last trip to Saudi Arabia, and then applying the algorithm above to its context to find a sightseeing event.

```
resolving references ...
The question is:
((<=> (sM-SIGHTSEE ACTOR HUM1)) MODE (*?*) TIME G0699)

The question type is "verification"
The question concept is:
((<=> (sM-SIGHTSEE ACTOR HUM1)))

   inferring a diplomatic trip

answering question using time context
 directly

The answer is:
((<=> (sM-SIGHTSEE ACTOR HUM1 SIGHT LOC23))
 PLACE POL13 TIME G0748)
Modality YES

 yes, at an oilfield in Dharan.
```

The diplomatic trip inferred by CYRUS while answering the question is the trip implied by "last time Vance was in Saudi Arabia". After retrieving that trip, its sequence of events is searched for a sightseeing episode.

8.2.7 Formulating an answer

After retrieving a target event from memory, there is often additional processing that must go on to formulate an answer. Consider, for example, the following question:

(Q8-1) Why did Vance meet with Gromyko yesterday?

In answering this question, it is necessary to first retrieve the target event "Vance's meeting with Gromyko yesterday", and then to figure out why it took place.

The processing after retrieving a target event from memory depends on the conceptual category the original question fit into. The conceptual categories CYRUS uses are listed below. They are based on those proposed by Lehnert (1978).

CYRUS' Question Categories

1. Identification	7. Place
2. Feature specification	8. Verification
3. Enablement	9. Duration
4. Instrumental/Procedural	10. Motivational
5. Concept completion	11. Result orientation
6. Time	

Figure 8-6

To answer an enablement question, for example, the enablement conditions are extracted from the events retrieved from memory. If there are no specific enablement conditions found, the instantiation strategy "Instantiate-Enablements" is used to infer the enabling conditions. CYRUS does that to answer the following question:

```
>How did Vance become Secretary of State?

The question is:
((CON (*?*) ENABLE
   ((ACTOR HUM1 IS (OCCUPATION VAL (RT-SEC-OF-STATE))))))
The question type is "enablement"
The question concept is:
((ACTOR HUM1 IS (OCCUPATION VAL (RT-SEC-OF-STATE))))

    applying Instantiate-Enablements to RT-SEC-OF-STATE

The answer is:
((ACTOR HUM1 IS (OCCUPATION VAL (RT-SEC-OF-STATE))))
Enablements:
 ((<=> ($APPOINT ACTOR HUM12 APPOINTEE HUM12)))

  He was appointed by President Carter.
```

In general, those questions which ask about content frame properties can be answered by applying associated context instantiation rules if actual answers are not available. "Result orientation" questions can be answered by applying "Instantiate-Results", and motivational questions can be answered by applying "Instantiate-Reasons".

Questions which ask about role fillers or slots of an episode can be answered by extracting the appropriate role filler from the retrieved events, or by using component instantiation strategies to infer what its value would be. "Time", "place", "duration", and "concept completion" questions are the question types which require extraction of slot-filler information. When CYRUS answers the question "Where is Vance today?", a "place" question, it uses a component instantiation strategy to infer that Vance's location must be his place of residence if it does not know of a trip he is on today. For a more complete explanation of the question categories CYRUS uses see Kolodner (1978).

After formulating an answer to a question, an English language generator called PHLUENT (McGuire, 1980) generates an English language answer. The generator was written by Rod McGuire, and transforms conceptual dependency representations of events into English. The generator was designed so that when the same event is used to answer more than one question, information given previously is not repeated in the later answer unless it is necessary to form a coherent sentence.

8.3 Memory update in CYRUS

CYRUS takes as input conceptual representations of episodes. Thus, they must be analyzed and a representation must be built before sending them to CYRUS. CYRUS has two modes of receiving representations of stories. In one mode, the stories are analyzed and their representations are coded by a human reader. The representations encoded by the human reader are integrated into CYRUS' memory.

In its second input mode, CYRUS is hooked up to FRUMP (DeJong, 1979), a computer program which reads and summarizes news stories from the UPI wire. The complete FRUMP-CYRUS hookup is called CyFr. FRUMP and CYRUS are run in conjunction every morning on stories about Vance and CYRUS that have come over the UPI wire since the last time it was run. When FRUMP understands one of those stories, it sends a conceptual representation of the story to CYRECV, a program which interfaces between FRUMP and CYRUS.

CYRECV converts FRUMP's representations into representations CYRUS can deal with and fills in contextual details. If, for example, FRUMP reports that Vance attended a meeting with Begin in Israel, then CYRECV will infer that Vance was on a diplomatic trip to Israel, that there were negotiations going on involving Israel and the United States, and that those negotiations were the goal of Vance's trip. CYRECV then sends its filled out event representations to CYRUS, which integrates them into memory as described in chapters 6 and 7. CYRUS can then answer questions about the new data.

Although individually CYRUS and FRUMP have been in operation for over two years, they have only been combined recently. At this writing, CyFr has processed approximately 50 stories about Vance and approximately 30 more about Muskie.

8.3.1 The FRUMP-CYRUS interface

In order to hook FRUMP and CYRUS together, there were a number of augmentations that had to go into both programs. FRUMP uses a knowledge structure called a <u>sketchy script</u> to hold the knowledge it needs to understand stories. While FRUMP originally had a sketchy script for meetings and diplomatic trips, it had no scripts corresponding to any of the other activities a Secretary of State does. Thus, the knowledge necessary for reading stories about press conferences, speeches, summit conferences, state dinners, welcomes, etc. had to be added. In addition, because information about Vance and Muskie is usually buried deep within stories about other topics, and because FRUMP was designed only to skim stories, FRUMP's story processing had to be augmented to allow it to read stories in more detail than it had been doing previously.

Before FRUMP and CYRUS were hooked up, CYRUS had been accepting hand-coded input. There were additional problems that had not been encountered in that mode that had to be taken care of when CYRUS began using FRUMP's summaries. The hand-coded representations for events that

CYRUS had been receiving had two characteristics that the FRUMP-coded stories did not have. First, they always specified the E-MOPs an event was an instance of. If a story stated that "Vance and Gromyko talked yesterday", the story was coded as "a diplomatic meeting between Vance and Gromyko". FRUMP, however, does not make inferences such as that. FRUMP's representations, however, are of low-level events, and use conceptual dependency primitives rather than specifying scripts or E-MOPs.

The second difference between the people-coded and FRUMP-coded representations of stories was that FRUMP does not make all the inferences about related events that people make while reading stories. When a person reads that "Vance and Gromyko met yesterday in Moscow to discuss SALT", he infers that Vance was on a diplomatic trip to Russia and that this meeting was part of the SALT negotiations. FRUMP, however, does not make those inferences.

A program had to be written to make these two types of inferences. FRUMP's conceptual dependency representations had to be turned into references to E-MOPs, and other related episodes had to be inferred. Although it is CYRECV, the program which interfaces CYRUS and FRUMP, which makes these inferences, it uses the same knowledge which CYRUS uses to construct contexts for search. Thus, CYRECV includes both types of context instantiation rules : those which construct contexts from components, and those which construct related contexts.

The following is an example of CyFr's operation.

```
@LFRUMP

FILE (JUN069 . J1) SKIMMED AT 8:52AM ON 6-13-1980

INPUT:  a085 r i ss czc ryr wyd
     Pm-vance 6-1------------By Anthony luke---madrid, spain
(upi)-secretary of State Cyrus Vance arrived today for talks
with Spanish leaders aimed at keeping this strategic
Mediterranean nation in the Western defense orbit.
     Vance flew to Madrid from the Netherlands where he
reported on the Salt ii agreement to a meeting of foreign
ministers of the North Atlantic Treaty Organization.
Foreign Minister Marcelino Oreja Aguirre left a cabinet
meeting to greet Vance at Barajas airport where the
secretary of state's plane landed at 12:30 p. m. (6:30
a. m. Edt), a half hour behind schedule.
     Vance greeted Oreja warmly. After a brief welcoming
ceremony he left with his wife in a 14-car motorcade for the
U. s. embassy. ...
     ----Upi 6-1 10:000 aed----***
```

```
SELECTED SKETCHY SCRIPT $MEET

REQUESTS:
 RQ154
  SATISFIED = T
     ((ACTOR (*VANCE*) <=> (*PTRANS*) OBJECT (*VANCE*)
                       TO (*MADRID*)
      TIME (*TODAY*) TENSE (*PAST*) GOAL &GOAL1)
  GOAL1
    ((ACTOR (*VANCE*) <=> (*MTRANS*) TO GROUP1)
     MANNER (*BILATERAL*) MEET-TYPE (*TALKS*))
 RQ155
  SATISFIED = T
     ((ACTOR (*VANCE*) <=> (*PTRANS*) OBJECT (*VANCE*)
                       TO (*MADRID*) FROM (*NETHERLANDS*))
      INST (*PLANE*) TENSE (*PAST*))
 RQ158
  SATISFIED = T
     ((ACTOR (*ORIJA*) <=> (*MTRANS*)
                       MOBJECT (*CONCEPT* TYPE (*GREETING*))
                       TO (*VANCE*))
       LOC (*MADRID*) TENSE (*PAST*))

CPU TIME FOR UNDERSTANDING = 91472 MILLISECONDS

** Generating
ENGLISH:
     Vance went to Madrid today to talk with Spanish
leaders.  Vance went on a plane from the Netherlands to
Madrid. Foreign Minister Marcelino Oreja Aguirre welcomed
Vance in Madrid.

Sending to CYRUS (DUMPS:  (JUN069 .  C1))
```

FRUMP's representations of these events are Conceptual Dependency representations and do not specify the E-MOPs they fall into. Thus, the first thing CYRECV does is to infer the E-MOPs these events would be part of. CYRUS knows that a *PTRANS* on a plane is a $FLY and constructs that context. FRUMP's MEET-TYPE in the MTRANS representation above specifies the context for the MTRANS. FRUMP'S "*TALKS*" corresponds to CYRUS' $MEET script. An MTRANS of a concept of type "*GREETING*" is recognized by CYRUS as a $WELCOME. CYRECV constructs those contexts also.

```
@CYRECV

reading in file ((CYRUS UPDATES) (JUN069 . C1)) from FRUMP
converting file to CYRUS format

inferring scenes from primitive acts
  inferring fly from *PTRANS*
  constructing $WELCOME from *MTRANS*
  constructing $MEET from *MTRANS*
```

CYRECV goes on to infer that if Vance flew from the Netherlands to Madrid, then he must have been on diplomatic trips to both places. It constructs representations for both of those diplomatic trip episodes, and puts the flight into their sequence of events. It also infers that Vance's talks with Spanish leaders must have been part of negotiations of some sort, that they must have been part of the diplomatic trip to Madrid, and that the welcome must have also been part of the trip.

```
inferring higher level episodes from scenes
  inferring sM-VIPVISIT from $FLY
    adding $FLY to its sequence of events
  inferring sM-VIPVISIT from $FLY
    adding $FLY to its sequence of events
  inferring I-NEGOTIATE from $MEET

  adding $WELCOME to the sequence of events of sM-VIPVISIT
  adding $MEET to the sequence of events of sM-VIPVISIT

Conversion complete.
Sending file ((CYRUS UPDATES) (JUN069 . F1)) to CYRUS
```

The following event representations are what CYRECV constructed at this point.

```
CY84:  ((<=> (sM-VIPVISIT ACTOR HUM1 DESTINATION POL45))
       TIME TIM100129)
  sequence of events: (CY79)

CY82:  ((<=> (sM-VIPVISIT ACTOR HUM1 DESTINATION POLN13
                         ORIGIN POL45 GOAL CY78))
       TIME TIM100127)
  sequence of events:  (CY79 CY81 CY78)

CY91:  ((<=> (I-NEGOTIATE ACTOR HUM1 OTHERS GROUPN45))
       TIME TIM100126)

CY78:  ((<=> ($MEET ACTOR HUM1 OTHERS GROUPN45))
       TIME TIM100126)
  contained in:  (CY82 CY91)

CY79:  ((<=> ($FLY ACTOR HUM1 DESTINATION POLN13 ORIGIN POL45))
       TIME TIM100122)
   contained in:  (CY84 CY82)

CY81:  ((<=> ($WELCOME ACTOR HUMN26 GUEST HUM1))
       PLACE POLN13 TIME TIM100120)
   contained in:  (CY82)
```

These representations are what is sent to CYRUS for integration into memory.

The following is a story CyFr has processed about Muskie.

```
@LFRUMP

FILE (JUN030 . M06) SKIMMED AT 4:1AM ON 6-13-1980

INPUT:  6-3
     Washington (Upi)-The State Department announced That
Secretary of State Edmund Muskie will travel to Asia this
month to attend a meeting of the Asean nations.
     The announcement said that Muskie, who will attend a
Nato Council meeting in Ankara June 24-25 will then fly to
Kuala Lumpur, Malaysia, to meet with other foreign ministers
of the alliance of non-communist Southeast Asian nations.
     Muskie is beginning the trip by accompanying President
Carter to Italy and Yugoslavia for state visits and to
Venice for the annual economic summit meeting of major
Western nations.
     Muskie is scheduled to return to Washington on June 29.
```

```
SELECTED SKETCHY SCRIPT $MEET

REQUESTS:
 RQ215
  SATISFIED = T
     ((ACTOR (*CARTER*) <=> (*PTRANS*) OBJECT (*CARTER*)
                          TO (*ITALY* *YUGOSLAVIA*)
                          FROM (*USA*))
      GOAL GOAL1)
  GOAL1 ((ACTOR (*CARTER*) <=> (*MTRANS*))
         MEET-TYPE (*TALKS*))

 RQ217
  SATISFIED = T
     ((ACTOR (*MUSKIE*) <=> (*PTRANS*) OBJECT (*MUSKIE*)
                          TO (*ASIA*) FROM (*USA*))
      TIME (*JUNE*) TENSE (*FUTURE*) GOAL GOAL2)
   GOAL2 ((ACTOR (*MUSKIE*) <=> (*MTRANS*) TO (*ASEAN*))
          MANNER (*BILATERAL*) MEET-TYPE (*TALKS*))

 ***BUNDLES: |CYRUS

REQUESTS:
 RQ325
  SATISFIED = T
     ((ACTOR (*MUSKIE*) <=> (*MTRANS*) TO (*NATO*))
      MANNER (*BILATERAL*) MEET-TYPE (*TALKS*) TIME (*JUNE*)
      LOC (*ANKARA*) TENSE (*FUTURE*))

CPU TIME FOR UNDERSTANDING = 63776 MILLISECONDS
** Generating ...
ENGLISH:
     Carter begins going from the United States to Italy and
Yugoslavia to talk. Secretary of State Edmund Muskie will
go from the United States to Asia this month to have talks
with ASEAN. Muskie will have talks with NATO in Ankara in
June.

***************

@CYRECV

reading in file ((CYRUS UPDATES) (JUN030 . E06)) from FRUMP

converting file to CYRUS format

inferring scenes from primitive acts
  inferring travel from *PTRANS*
  inferring travel from *PTRANS*
  constructing $MEET from *MTRANS*
  constructing $MEET from *MTRANS*
```

```
inferring higher level episodes from scenes
  inferring I-NEGOTIATE from $MEET
    adding $MEET to its sequence of events
  inferring sM-VIPVISIT from sM-TRAVEL
    adding sM-TRAVEL to its sequence of events
  inferring sM-VIPVISIT from sM-TRAVEL
    adding sM-TRAVEL to its sequence of events
  inferring sM-VIPVISIT from $MEET
    adding $MEET to its sequence of events

Conversion complete.
Sending file ((CYRUS UPDATES) (JUN030 . E06)) to CYRUS
```

8.3.2 Integrating events from FRUMP into memory

In CYRUS' normal mode of operation, it integrates events into memory as described in chapters 6 and 7, with one variation. CYRUS does not index in newly-created sub-MOPs until the E-MOP has reached a reasonable size. In CYRUS, that number is 6. Until then, it keeps a list of the events in the E-MOP. There are two reasons for this. First, that keeps memory from growing too fast. Second, it takes that many events before an E-MOP's generalized *information stabilizes*. Waiting until that time to index events in the E-MOP means that better generalizations can be made the first time.

Some additional updating problems had to be taken care of in integrating stories processed by FRUMP into memory. Before FRUMP and CYRUS were hooked together, all of its inputs had been hand-processed. Because of that, the same event was never added to CYRUS' memory more than once. CYRUS did not have the capability of recognizing if an event was already in memory. The same event is described in many different news stories, however, all of which are sent to CYRUS. Thus, there is a need to recognize whether or not an event is already in memory.

Because a description of an event from a news story might have additional or different information than an event already in memory, but still refer to the same event, simple traversal treating the new event as the target event is not sufficient. Consider, for example, the following two descriptions of the same event:

EV8-1: Vance went to Russia last week.

EV8-2: Vance went to Russia last week to negotiate SALT.

Because the second event has more information than the first, it cannot be used as a target event to retrieve EV8-1. Thus, if EV8-1 were already in memory and EV8-2 were added later, it would be added as a separate event. What CYRUS does to get around that problem is to query memory with the smallest amount of information that will describe an event.

In the case of events that take place over a long period of time, that includes the actor, the time, and the features that CYRUS has marked as identifying. Since its place identifies a trip, CYRUS would search memory using the specification "Vance went to Russia last week" to see if EV8-2 were already in memory. If EV8-1 were already in memory, CYRUS would find it and merge the information in the two events.

For events that normally take place over a small period of time (a few hours or less), CYRUS uses the entire given specification to search memory. This method is not foolproof. If the following two descriptions were of the same event, CYRUS would not recognize that fact, and would add each one to memory separately.

EV8-3: Vance met with Gromyko in Moscow yesterday.

EV8-4: Vance attended a meeting about SALT yesterday.

8.4 A Vance example

Upon receiving representations of events to be added to memory, CYRUS checks to see if an event is already in memory and then integrates it into memory using the procedures described in chapters 6 and 7. The following output is CYRUS′ integration into memory of the Vance story presented in section 8.4.1. FRUMP′S summary of that story is repeated below.

> Vance went to Madrid today to talk with Spanish leaders. Vance went on a plane from the Netherlands to Madrid. Foreign Minister Marcelino Oreja Aguirre welcomed Vance in Madrid.

Figure 8-7

When CYRUS adds this episode to its memory, it adds each of the individual events to the E-MOPs it belongs in. Thus, it adds a trip to Madrid, one to the Netherlands, a diplomatic meeting, a welcome, a flight, and negotiations. It first checks to see if any of the events are already in memory. If so, it updates the information it has about those events.

```
@CYRUS

Are you interested in Muskie or Vance?  (M or V) : *VANCE

*(PROCESS-FILES)

reading in file (JUN069 . F1)
updating memory with new events

searching memory for CY84 -- sM-VIPVISIT ...   not found
searching memory for CY79 -- $FLY ...   not found
searching memory for CY82 -- sM-VIPVISIT ...   not found
searching memory for CY81 -- $WELCOME ...   not found
searching memory for CY78 -- $MEET ...   not found
searching memory for CY91 -- I-NEGOTIATE ...   not found
```

In this case, CYRUS did not have previous knowledge about any of the events in this story. It goes on to add each event to memory.

```
Adding item ((<=> (I-NEGOTIATE ACTOR HUM1 OTHERS GROUPN45))
             TIM100126)
    creating index for (PARTICIPANTS NATIONALITY) = POLN3
    creating index for PARTICIPANTS = GROUPN45
13022 msec CPU (1714 msec GC),20000 msec clock,25593 conses
NODES = 2226 ALISTS = 576
```

The first event it adds to memory is the negotiating episode that it had previously inferred. It creates indices for the participants -- a group of Spanish leaders (GROUPN45), and for the nationality of the participants -- Spanish (POLN3). Because it had no other events with either of those features, this negotiating episode is now uniquely specified in memory in two ways. CYRUS goes on to add the next event.

```
Adding item ((<=> ($MEET ACTOR HUM1 OTHERS GROUPN45))
             TIME TIM100126)
  it is part of (sM-VIPVISIT I-NEGOTIATE) episodes
     creating index for
         (PARTICIPANTS NATIONALITY) = POLN3 in $MEET
     creating index for PARTICIPANTS = GROUPN45 in $MEET
     creating index for PLACES = POLN3 in $MEET
     creating index for PLACES = POLN13 in $MEET
11616 msec CPU (1895 msec GC),33000 msec clock,54688 conses
NODES = 2231 ALISTS = 576
```

In adding the meeting with Spanish leaders to memory, CYRUS indexes it in the "diplomatic meetings" MOP ($MEET) according to the nationality of its participants, the participants themselves, the country it took place in, and the city. Again, none of these features correspond to features of other events in memory, so the meeting is indexed uniquely by each of these indices.

```
Adding item ((<=> ($WELCOME ACTOR HUMN26 GUEST HUM1))
             PLACE POLN13 TIME TIM100120)
  it is part of (sM-VIPVISIT sM-VIPVISIT) episodes
Reminded of ((<=> ($WELCOME ACTOR HUM61 GUEST HUM1))
             PLACE LOC20 TIME TIM403)
  because in both  Vance does $WELCOME with
      (TIME SEASON) = SUMMER
     creating new MOP
        no additional similarities to generalize on
     creating index for ACTOR = HUMN26 in $WELCOME
     creating index for PLACES = POLN3 in $WELCOME
     creating index for PLACES = POLN13 in $WELCOME
844 msec CPU (0 msec GC), 1000 msec clock, 2494 conses
NODES = 2234 ALISTS = 576
```

When CYRUS adds the "welcome" to its memory, it is reminded of a previous welcoming ceremony that also happened in the summer. This reminding happens because the feature "time is summer" already indexes another event. CYRUS thus creates a new E-MOP indexed by that feature, and attempts to make generalizations about "welcoming ceremonies that take place during the summer". In fact, it can find no domain-relevant similarities between them. If, after other summer welcoming ceremonies are added to memory, it can find no similarities between them, it will mark "time is summer" as a non-predictive index, and it will no longer index welcomes by that feature. The other indices CYRUS creates for this welcoming ceremony correspond to the person who did the welcoming -- Oreja (HUMN26) -- and its location.

```
Adding item ((<=> (sM-VIPVISIT ACTOR HUM1 DESTINATION POLN13
                               ORIGIN POL45))
             TIME TIM100127)
  it is part of (I-NEGOTIATE) episodes
  its seq of events includes ($FLY $WELCOME $MEET)
   Putting this event into MOP with specifications
      Vance does sM-VIPVISIT with (TIME SEASON) = SUMMER
      assuming that
       GOALS = I-NEGOTIATE
   creating index for ORIGIN = POL45 in sM-VIPVISIT
   creating index for PLACES = POLN3 in sM-VIPVISIT
   creating index for PLACES = POLN13 in sM-VIPVISIT
   Putting this event into MOP with specifications
       Vance does sM-VIPVISIT with PLACES = LOC31
      assuming that
       (EVENTS MOP) = $REPORT
       GOALS = I-NEGOTIATE
   creating index for
         (EVENTS PARTICIPANTS) = GROUPN45 in sM-VIPVISIT
11825 msec CPU (2012 msec GC),50000 msec clock,47891 conses
NODES = 2240 ALISTS = 576
```

CYRUS next adds the trip to Madrid (POLN13) to its memory. The first feature it chooses for indexing is the time -- summer. In this case, CYRUS already has an E-MOP for "diplomatic trips in the summer", and it adds the trip to that E-MOP. Apparently, that E-MOP holds the generalized information that "diplomatic trips in the summer have the goal of negotiations". It thus infers that this trip was also for that purpose. CYRUS then goes on to index this trip as a "trip from the Netherlands", a "trip to Spain", a "trip to Madrid", a "trip to Europe", and a "trip whose events had Spanish leaders as participants". In indexing the trip as a "trip to Europe" (LOC31), CYRUS finds that it already has an E-MOP for "trips to Europe", and it adds this event to that E-MOP. It makes the generalization that the trip was for the purpose of negotiations, and that it included a press conference ($REPORT), since those are features it has generalized about "trips to Europe". The other indexed features caused no reminding, nor did it they retrieve E-MOPs. Those features, then, uniquely index this trip.

The flight to Madrid is added to memory similarly.

```
Adding item ((<=> ($FLY ACTOR HUM1 DESTINATION POLN13
                                   ORIGIN POL45))
             TIME TIM100122)
  it is part of (sM-VIPVISIT sM-VIPVISIT) episodes
     creating index for (TIME SEASON) = SUMMER in $FLY
     creating index for ORIGIN = POL45 in $FLY
     creating index for (ORIGIN PLACE) = LOC31 in $FLY
     creating index for PLACES = POLN3 in $FLY
     creating index for PLACES = POLN13 in $FLY
1709 msec CPU (0 msec GC), 5000 msec clock, 6634 conses
NODES = 2245 ALISTS = 576
```

in adding the last event from this story to memory, the trip to the Netherlands (POL45), the size of its E-MOP "trips to Europe" reaches its critical size (6). CYRUS thus refines the generalizations it has for that E-MOP and begins indexing its events.

```
Adding item ((<=> (sM-VIPVISIT ACTOR HUM1 DESTINATION
POL45))
             TIME TIM100129)
  its seq of events includes ($WELCOME $FLY)
 Putting this event into MOP with specifications
     Vance does sM-VIPVISIT with (TIME SEASON) = SUMMER
    assuming that
     GOALS = I-NEGOTIATE
 Putting this event into MOP with specifications
     Vance does sM-VIPVISIT with PLACES = LOC31
    assuming that
     (EVENTS MOP) = $REPORT
     GOALS = I-NEGOTIATE
    no additional similarities to generalize on
    removing generalizations
        (EVENTS MOP) = $REPORT
    from MOP  Vance does sM-VIPVISIT with PLACES = LOC31
 creating index for (TIME SEASON) = SUMMER in
    MOP  Vance does sM-VIPVISIT with PLACES = LOC31
 creating index for PLACES = POL45 in
    MOP  Vance does sM-VIPVISIT with PLACES = LOC31
14948 msec CPU (2059 msec GC),29000 msec clock,34836 conses
NODES = 2283 ALISTS = 584

updating complete
```

The first index CYRUS chooses is an index for time -- summer. It adds this event to the E-MOP "diplomatic trips in the summer" similarly to the way it added the trip to Madrid to that E-MOP. It then adds this trip to its E-MOP "trips to Europe" (LOC31). As in adding the trip to Spain, it makes the generalizations that the trip to the Netherlands included a press conference and had the goals of negotiations.

Now that it has a reasonable number of trips to Europe indexed, it refines the generalizations it has made about "trips to Europe", and also starts indexing within that E-MOP. It can make no additional generalizations beyond the ones it already has. After examining the generalizations it does have, it decides that the fact that there was a press conference during the trip was a bad generalization. It thus deletes that from its generalizations and builds an index for it. It indexes this trip in the "diplomatic trips to Europe" MOP as a "trip in the summer" and as a "trip to the Netherlands".

After integrating these events into memory, CYRUS answers the following questions. Note that CYRUS' parser can parse questions asked of it in the second person or third person. When the Muskie data base is being queried, the word "you" is equivalent to "Muskie". In the questions below, Muskie is referred to as "you".

```
*(DIALOG2)

Enter next question

>WERE YOU IN EUROPE LAST YEAR?

The question is:
((ACTOR HUM1 IS (*LOC* VAL LOC31)) TIME G0717 MODE (*?*))

searching memory for question concept
  searching directly for input -- sM-VIPVISIT
   found (CY82)

 yes, most recently in Madrid.

Enter next question

>WERE YOU WELCOMED THERE?

The question is:
((<=> ($WELCOME GUEST HUM1) MODE (*?*)
 TIME G1831 PLACE POLN3)

answering question using previous context:
 directly

 yes, by Foreign Minister Marcelino Oreja Aguirre.
```

```
Enter next question

>WHO DID YOU TALK TO THERE?

The question is:
((ACTOR HUM1 <=> (*MTRANS*) TO (*?*))
 TIME G1519 PLACE POLN3)

   inferring undifferentiated political meeting

answering question using previous context:
 directly

 a group of Spanish officials.

Enter next question

>DID YOU ALSO GO TO HOLLAND?

The question is:
((ACTOR HUM1 <=> (*PTRANS*) OBJECT HUM1 TO POL45)
 TIME G1607 MODE (*?*))

answering question using previous context:
 directly
 using top-level events
 using co-occuring events
current context not applicable
searching memory for question concept
  searching directly for sM-VIPVISIT
   found (CY84)

 yes.
```

8.5 A Muskie example

The entire Muskie data base was built up from stories processed by FRUMP. The events that were added to the Vance data base in the example above were added to a detailed data base which already had a lot of events in it. The Muskie data base, on the other hand, has many fewer events in it, and those that it has are more sparse. Because of the sparseness of the data, CYRUS often cannot make any generalizations when it is reminded.

The following is the output when the Muskie story above is added to the Muskie data base. Recall that FRUMP produced the following summary:

Carter begins going from the United States to Italy and Yugoslavia to talk. Secretary of State Edmund Muskie will go from the United States to Asia this month to have talks with ASEAN. Muskie will have talks with NATO in Ankara in June.

The indices CYRUS *creates as* the new events are added have not been output in this run. However, all of the reminding and subsequent generalization that CYRUS does in adding this episode are illustrated.

```
@CYRUS

Are you interested in Muskie or Vance?  (M or V) : *MUSKIE

*(PROCESS-FILES)

reading in file (JUN030 . F06)
updating memory with new events

searching memory for CY77 -- sM-VIPVISIT ...   not found
searching memory for CY71 -- sM-TRAVEL ...   not found
searching memory for CY73 -- $MEET ...   not found
searching memory for CY70 -- sM-TRAVEL ...   not found
searching memory for CY76 -- sM-VIPVISIT ...   not found
searching memory for CY74 -- I-NEGOTIATE ...   not found

Adding item ((<=> (I-NEGOTIATE ACTOR HUM0 OTHERS ORG16))
             PLACE POLN12 TIME TIM100115)
  its seq of events includes ($MEET)
181 msec CPU (0 msec GC), 0 msec clock, 662 conses
NODES = 108 ALISTS = 48

Adding item
   ((<=> (sM-VIPVISIT ACTOR HUM0 DESTINATION LOCN6
                      ORIGIN POL1 GOAL
                      ((<=> ($MEET ACTOR HUM0 OTHERS ORGN9))
                       TIME TIM100114)))
     TIME TIM100117)
 to memory
  its seq of events includes (sM-TRAVEL sM-TRAVEL $MEET)
Reminded of
    ((<=> (sM-VIPVISIT ACTOR HUM0 DESTINATION LOC31
                       ORIGIN POL2 GOAL
                       ((<=> ($MEET ACTOR HUM0 OTHERS HUM66))
                        TIME TIM10028)))
      TIME TIM1003)
because in both  Muskie does sM-VIPVISIT with
    GOALS = ($MEET)
   creating new MOP
   no additional similarities to generalize on
```

```
Reminded of
    ((<=> (sM-VIPVISIT ACTOR HUMO DESTINATION LOC31
                       ORIGIN POL2 GOAL
                       ((<=> ($MEET ACTOR HUMO OTHERS HUM66))
                        TIME TIM10028)))
     TIME TIM1003)
because in both  Muskie does sM-VIPVISIT with
    ORIGIN = POL1
   creating new MOP
generalizing that when Muskie does sM-VIPVISIT with
    ORIGIN = POL1
often
    GOALS = ($MEET)
1150 msec CPU (0 msec GC), 2000 msec clock, 3675 conses
NODES = 109 ALISTS = 48

Adding item ((<=> (sM-TRAVEL ACTOR HUMO DESTINATION LOCN6
                              ORIGIN POL1)) TIME TIM100102)
  it is part of (sM-VIPVISIT sM-VIPVISIT) episodes
Reminded of ((<=> (sM-TRAVEL ACTOR HUMO DESTINATION LOC31
                              ORIGIN POL1)) TIME TIM1002)
    because in both  Muskie does sM-TRAVEL with
    ORIGIN = POL1
   creating new MOP
    no additional similarities to generalize on
712 msec CPU (0 msec GC), 1000 msec clock, 2482 conses
NODES = 110 ALISTS = 48

Adding item ((<=> ($MEET ACTOR HUMO OTHERS ORG16))
             TIME TIM100110 PLACE POLN12)
  it is part of (I-NEGOTIATE sM-VIPVISIT) episodes
217 msec CPU (0 msec GC), 0 msec clock, 810 conses
NODES = 113 ALISTS = 48

Adding item ((<=> (sM-VIPVISIT ACTOR HUMO DESTINATION
POLN12))
             TIME TIM100118)
  its seq of events includes (sM-TRAVEL $MEET sM-TRAVEL)
  it co-occurs with (sM-VIPVISIT I-NEGOTIATE)
132 msec CPU (0 msec GC), 1000 msec clock, 521 conses
NODES = 115 ALISTS = 48

updating complete
```

After adding this episode to memory, CYRUS answers the following questions about it. As in the Muskie examples above, the data base is queried in the second person. When the Vance data base is being queried, CYRUS knows that the word "you" refers to "Vance".

```
*(DIALOG2)

Enter next question

>Have you been to Europe recently?

The question is:
((ACTOR HUMO IS (*LOC* VAL LOC31)) TIME G0668 MODE (*?*))
searching memory for question concept
  searching directly for sM-VIPVISIT
   found (CY2)

 yes, most recently last month.

Enter next question

>Why did you go there?

The question is:
((CON (*?*) REASON
   ((ACTOR HUMO <=> (*PTRANS*) OBJECT HUMO TO LOC31)
    TIME G0790)))
answering question using previous context:
 directly

 to talk to Andrei Gromyko.

Enter next question

>Who did you talk to there?

The question is:
((ACTOR HUMO <=> (*MTRANS*) TO (*?*))
 TIME G0896 PLACE LOC31)

   inferring undifferentiated political meeting

answering question using previous context:
 directly

 to NATO in Brussels on MAY 14 1980 and to Andrei
 Gromyko in Vienna.
```

```
Enter next question

>Are you going to Asia?

The question is:
((ACTOR HUMO <=> (*PTRANS*) OBJECT HUMO TO LOCN6)
 MODE (*?*))

searching memory for question concept
  searching directly for sM-VIPVISIT
   found (CY76)

 yes, this month.

Enter next question

>Who will you talk to?

The question is:
((ACTOR HUMO <=> (*MTRANS*) TO (*?*)) TIME G1136)

   inferring undifferentiated political meeting

answering question using previous context:
 directly

 with NATO in Ankara, Turkey.
```

8.6 Differences between the Vance and Muskie memories

The Vance and the Muskie memories start out the same, but after adding events to the two data bases, their organizations are quite different. The following two E-MOPs, for example, are the Vance and Muskie "diplomatic meetings" MOPs, each with 10 meetings indexed in it:

```
        Vance's "diplomatic meetings" E-MOP -- $MEET

content frame: included in "negotiations"
               participants are foreign diplomats
               topic is an international contract
               topic involves the United States
               is political and occupational for Vance
               topic is Arab-Israeli peace
               underlying topic is peace
               involves Israel and the Arabs
           /   |          |           |    |      \
diffs:    /    |          |           |    |       \
         /     |     participants     |     \       \
      topic    |     nationality      |   included   \
     /  \      |       /  |   |  \    |     in        \
    /   Salt   | Israel   |   | USSR  |      |         |
Camp     |     | (MOP10)  |   |   |   |    summit      |
David   EV2    |       Egypt  | EV2   |    (MOP9)      |
Accords        |      (MOP6)  |       |          underlying
(MOP5)         |           Jordan     |              topic
               |             |        |                |
               |            EV6       |                |
             topic's                  |               arms
              sides             participants'        limits
             /   \              occupations            |
            /    USSR          /     |   \            EV2
         Egypt     |       foreign   |    def min
         & Israel EV2      minister  |       |
         (MOP8)            (MOP7)   head    EV5
                                  of state
                                   (MOP3)
```

Figure 8-8

```
          Muskie's "diplomatic meetings" E-MOP -- $MEET

content frame:  included in "negotiations"
                participants are foreign diplomats
                topic is an international contract
                topic involves the United States
                is political and occupational for Muskie
                  /           |
diffs:           /            |
           underlying    participants       participants'
             topic       nationality         occupations
               |           /  |    | \            |
               |     Israel   |    | Thailand  foreign
          military     |   England |    |      minister
             aid      EV2     |    |   EV3      (MOP2)
              |              EV4   |
             EV1                  USSR
                                 (MOP1)
```

Figure 8-9

Each of these E-MOPs shows all of the indices in "diplomatic meetings" ($MEET), except the indices for actual participants and those for location.

There are 4 obvious differences between these two E-MOPs.

1. The indices are different.

2. The types of indices are different. While the Vance E-MOP has topic indices and larger episode indices, the Muskie E-MOP has neither of those.

3. The content frames of the two E-MOPs are different.

4. The Vance E-MOP indexes mostly sub-MOPs, and the Muskie E-MOP indexes mostly individual events.

The differences between these two E-MOPs are representative of the differences between other E-MOPs in the two memories. Three factors contribute to the differences between them. First, the experiences the two men have had are different. This is the reason for the differences between the indices in the two E-MOPs. The events entered into the Vance memory included meetings with Egyptians, while those entered into the Muskie data base did not. Thus, the E-MOP from the Vance data base has indices for "participants are Russian", while the Muskie E-MOP doesn't. On the other hand, Muskie met with Thai representatives, and Vance did not. Thus, there is an index corresponding to "participants are Thai" in the Muskie E-MOP but not in the Vance one.

Second, the data entered into the Vance data base is much more detailed than that entered into the Muskie memory. The Vance data base holds mostly hand-coded stories, while the Muskie memory has FRUMP-coded stories. Because the topics of meetings are so unconstrained, FRUMP has difficulty picking up meeting topics. Thus, while a typical meeting Vance has a topic specification, Muskie's meetings do not. A typical meeting description entered into the Vance data base would be "a meeting with Begin about the Camp David Accords in Jerusalem", while a typical meeting Muskie had with Begin would be entered as a "diplomatic meeting with Begin in Jerusalem". This factor accounts for the differences in the types of indices in the two E-MOPs. Because the Muskie memory is not usually aware of the topics of Muskie's meetings, it cannot index them by aspects of their topics.

The third factor which accounts for differences between the two E-MOPs is the degree of similarity between the events entered into the E-MOPs. The first ten events added to the Vance E-MOP were very similar to each other. Eight of them were meetings about the Camp David Accords. On the other hand, except for three meetings with Gromyko the meetings entered into the Muskie data base had very different participants and locations.

This factor accounts for differences (3) and (4) above. The Vance "diplomatic meetings" MOP has a more filled out content frame because the meetings entered into it had so many similarities. The Vance E-MOP indexes more sub-MOPs than individual events for the same reason. The Muskie E-MOP indexes mostly individual events and not sub-MOPs because most of its events were dissimilar. *The extent of new category creation, then, is a function of the degree of similarity between the items added to an E-MOP*, and not on the number of items it organizes.

8.6.1 CYRUS' generalizations

There is a fifth difference between the two memories which does not show in the illustrations above. That difference is in the generalizations made by the two memories. When new categories are created, generalizations are abstracted from the similarities between its events. Thus, each sub-MOP represents a set of generalizations. If the extent of new category creation is dependent on the similarities between events, then the extent of generalization in memory is also dependent on the extent of the similarities between the items in memory.

Consider, first the Muskie "diplomatic meetings" MOP illustrated above. Its sub-MOPs include "meetings with Russians" and "meetings with foreign ministers". Its "meetings with Russians" MOP has three events indexed in it and has the following content frame.

Muskie's "meetings with Russians" E-MOP

content frame: participants are Russian
participant is a foreign minister
participants include Gromyko

Figure 8-10

One can read off the generalizations an E-MOP organizes as "when <E-MOP indices> then <content frame features which do not index the E-MOP>". Thus, this E-MOP organizes the following generalization:

GEN8-1: Meetings with Russians are with foreign ministers and include Gromyko.

Another sub-MOP that the Muskie "diplomatic meetings" MOP indexes is "meetings with foreign ministers". That E-MOP organizes three meetings with Gromyko and a meeting with the foreign minister of Israel, Itzhak Shamir. The meetings with Gromyko have similarities, but the only similarities between them and the meeting with Shamir are the meeting format. CYRUS thus does not have any generalizations associated with "meetings with foreign ministers", but is waiting until it has six meetings before it decides what they have in common. The generalizations it is considering are those in the content frame of "meetings with Russians" (i.e., the similarities between the majority of its meetings). After an additional two meetings are added to that E-MOP, CYRUS will either find similarities between them and fill in its content frame, or it will find no similarities and mark "participant is a foreign minister" as a non-predictive feature and will no longer index it.

The Vance "diplomatic meetings" MOP has many more sub-MOPs and thus many more generalizations. After ten meetings were added to that E-MOP, only one of them had been with a Russian. Thus, it did not have a sub-MOP for "meetings with Russians". After additional meetings with Gromyko were added to its memory, however, it did create that E-MOP. The following figure illustrates the Vance memory's "meetings with Russians".

Vance's "meetings with Russians" E-MOP

content frame: participants are Russian
participant is a foreign minister
participants include Gromyko
topic is SALT
topic is arms limitations
topic involves Russia
location is Moscow

Figure 8-11

Contrast this E-MOP to the Muskie "meetings with Russians" MOP in Figure 8-10. This E-MOP includes the generalized information that was included in the Muskie E-MOP, and in addition holds generalized information about the topics of these meetings and their locations. Thus, in addition to the generalization listed above, the Vance "meetings with Russians" MOP organizes the following two generalizations:

GEN8-2: Meetings with Russians are about SALT.

GEN8-3: Meetings with Russians are in Moscow.

The reason CYRUS can make the first generalization in the Vance data base and not in the Muskie one is that the events entered into the Vance memory have topic information. The generalizations that can be made depend on the amount of information available. As a result of its sparseness, the Muskie data base has many fewer indices for each event and also makes fewer generalizations.

The reason CYRUS makes the second generalization in the Vance data base and not in the Muskie data base is because Vance's meetings with Russians happened in Russia, while Muskie's did not have location in common. Generalizations in the two data bases correspond to the differing similarities between the two men's experiences.

The generalizations that CYRUS makes are correspondences between features of events. The following generalizations are representative of those CYRUS has made about Vance's activities.

CYRUS' generalizations

1. Meetings in the Middle East are about Arab-Israeli peace.

2. When Vance meets in the Middle East with Egyptians, he also meets with Israelis, and the topic is the Camp David Accords.

3. When Vance meets with Begin, the topic is the Camp David Accords.

4. When Vance goes to a conference concerning Egyptian concerns, the topic of the conference is the Camp David Accords.

5. When Vance negotiates with Egyptians, the topic is the Camp David Accords.

6. Trips to Russia are for the purpose of negotiating SALT.

7. When Vance attends a conference with Gromyko, the topic is SALT.

8. When Vance takes a trip to negotiate Arab-Israeli peace, the trip is to the Middle East.

Figure 8-12

Note that the first two generalizations both refer to "meetings in the Middle East", but the topics they predict are different. That is because the first generalization is a generalization about "meetings in the Middle East" in general. It is associated with that E-MOP. The second is a generalization associated with "meetings in the Middle East with Egyptians", a sub-MOP of "meetings in the Middle East" and "meetings with Egyptians". In fact, "the Camp David Accords" is a specialization of "Arab-Israeli peace", and the two do not conflict.

8.7 Summary

CYRUS searches and updates its memory as described in previous chapters. In addition, CYRUS′ question answering loop includes processes of parsing a question, resolving its pronominal references, extracting the target concept, and after searching its memory, formulating an answer. CYRUS updates its memory when it receives a story summary from FRUMP (DeJong, 1979). CYRUS′ Vance and Muskie data bases differ in level of detail and relative similarities between the events in the two data bases. Because the events in the Vance data base are more similar to each other, CYRUS has made more interesting generalizations concerning Vance′s activities than it has made about Muskie′s activities.

CHAPTER 9

Psychology and Computer Science Approaches to Memory

9.1 Introduction

CYRUS, and the theory of reconstructive memory which it models, have been presented as models of (1) conceptual memory, (2) long term memory for events, (3) question answering, (4) data base organization for large amounts of inter-related facts, and (5) intelligent information retrieval. Problems of memory organization, retrieval from memory, and memory update have been addressed. Each of these areas has been studied individually in the past by researchers in computer science and psychology. None, however, have addressed problems in each of these areas simultaneously, and treated them as the same problem. In this chapter, related work in each of these areas will be discussed.

9.2 Approaches to long-term memory

This research is certainly not the first research to claim that human long term memory is reconstructive. Psychologists as far back as Bartlett (1932) have described memory as reconstructive. Psychologists, however, have presented very little in the way of an overall theory of long term memory for episodes.

A number of recent experimental results illustrate the reconstructive nature of human memory. In asking people to name persons in their high school classes, Williams (1978) found that his subjects recalled features of a person's name, such as its first letter, the number of syllables, and what it rhymed with, and used that information to come up with the name. When he asked people to recall whether a karate expert in a particular story had broken a block, Spiro (1979) found that they used their knowledge about karate experts in general to answer the question, rather than retrieving actual story details. In both of these cases, only partial information was retrieved from memory, and using generalized knowledge, "actual" items were reconstructed.

The recent work of Norman and Bobrow (1975, 1977) addresses some of the problems addressed in this thesis. They have proposed a theory of memory "descriptions" and constructive processes that operate on those descriptions. Their theory is a theory of reconstructive memory -- in order to find something in memory, its description must first be constructed. They have described reconstruction as a process comprised of (1) specification of a retrieval description, (2) search for that description, (3) evaluation of the memory record retrieved, and (4) failure diagnosis of that evaluation resulting in a new retrieval specification.

While the reconstructive process they have described is similar in nature to that described in this thesis and used by CYRUS, Norman and Bobrow have chosen to ignore the problem of memory organization, and thus have not addressed the problem of what kinds of information are included in "descriptions" and how they might be organized with respect to each other in memory. Nor do they address the problem of how any particular specified description is found. Rather, their concern is with very general processes which describe a retrieval mechanism independent of any particular organization.

Williams (1978) work has drawn on the Norman and Bobrow theory of descriptions, and is closest to that presented in this thesis. Williams (1978) has described the reconstructive process as a three-step retrieval process: find a context, search, verify. Each of those steps, he explains, is a reconstructive processes with the same three steps. Williams has gone on to describe some of the strategies people use in constructing "descriptions" (Norman & Bobrow, 1977, Bobrow & Norman, 1975) necessary for remembering people they have known.

Although Williams identified some of the strategies people used for that task, his emphasis was not on identifying strategies for retrieval or explaining a memory organization. Like Norman and Bobrow, his interest was in identifying and explaining very general memory phenomena independent of a specified underlying memory organization. As a result, he has uncovered many of the general reasoning processes people use in remembering, and has explained some of the retrieval failures people have, but his explanations are on a very general level, and do not explain how particular pieces of knowledge can guide reconstructive processing. He observes that contexts for search are chosen, for example, but cannot describe how memory can guide that search.

In many ways, the work presented in this thesis is complementary to the work of Norman and Bobrow, and of Williams. While they have proposed general mechanisms for retrieval without worrying about underlying memory organization, this thesis has proposed a memory organization which supports reconstructive processing, and well-specified processes for retrieval which depend on that underlying organization of information and knowledge in memory.

Another theory in psychology which explains very low level memory mechanisms is the "spreading activation" (Anderson, 1976, Anderson and Bower, 1973) theory, or theory of "associative memory". According to this theory, memory is seen as a network of nodes, or associations. When a description of something or a series of features are entered into

memory, nodes representing those features are excited. This excitation is transferred to neighboring nodes, though not completely. In the simple model, excitation is additive. When paths of activation converge, their intersections will be more excited than the rest of memory. When the excitation of a node reaches a threshold, that node's contents emerge into conscious thought. This model has been used to describe many memory phenomena (e.g., Quillian (1968), Anderson (1976), Anderson and Bower (1973)).

Although this model seems to explain memory's organization, it does not attempt to explain what the nodes, links, or associations in the network look like. It is *not* a model of memory organization, then, but rather a model of very low level memory processing. This theory, then, like that of Norman and Bobrow, is complementary with that presented in this thesis. It is an attempt to explain low level traversal from node to node in memory. This thesis has presented a theory of the content of the nodes, their connections, and higher level processes guiding communication between the nodes.

A third theory of long-term memory, developed concurrently with this one, and complementary to it, is Schank's (1980) theory of Memory Organization Packets (MOPs). Schank's MOPs are an attempt to explain how personal experiential memories can be organized in memory. He proposes that MOPs organize experiences according to their differences and also organize similarities between the events. This work has extended that theory by explaining processes for index selection, memory organization maintenance, and retrieval.

The major differences between Schank's MOPs and the E-MOPs presented in this thesis are differences in emphasis. He was interested in explaining the interconnectness of memory structures, while this work has been more concerned with storage and retrieval of individual episodes.

9.3 Organization of conceptual memory

Unlike psychological research which has not focused on memory organization, previous research in A. I. has addressed problems of organizing concepts in memory. The emphasis in previous A. I. research on conceptual memory, however, has been with conceptual structures used for understanding, and not with structures used for long term storage of events. Thus, there has been no previous work done in A. I. on the organization of events in a long term memory. The emphasis in natural language research has been on text understanding and isolating the conceptual structures necessary for the understanding process, but not on storing the contexts of those texts for later retrieval or use in understanding. This research, on the other hand, has been concerned with how to organize the information extracted from texts in a long term memory, and how to retrieve it from that organization. The model of long term memory which has been presented is the logical next step in explaining what conceptual memory looks like, and follows naturally from previous A. I. work on conceptual memory.

Previous text understanding systems have focused on the use of various knowledge structures in understanding and controlling inference. SAM (Cullingford, 1977) and FRUMP (DeJong, 1979), for example, used scripts as knowledge structures which held the information necessary for understanding script-based stories. Charniak's (1977) frames hold the conceptual information necessary for understanding stories about painting. PAM (Wilensky, 1978) used goals and planboxes as knowledge structures which held the information necessary for understanding stories about goals. The BORIS system (Dyer and Lehnert, 1980) uses MOPs to hold the generalized information needed to understand more complicated stories.

There are two important differences between those systems and CYRUS. The first difference is one of emphasis. Each of those endeavors isolated a particular type of knowledge necessary for understanding, represented that information in conceptual structures, and used that knowledge in understanding, but none of these programs remembered stories they had read after processing them. CYRUS, on the other hand, is designed to store facts extracted from news stories in long term memory so that they can later be retrieved.

Another important difference between those systems and CYRUS, which follows from the first, is in the different ways conceptual structures are used. Understanding systems used conceptual structures as knowledge structures for organizing generalized information. By contrast, the conceptual structures described in this thesis not only organize generalized information, but also act as conceptual categories to organize what has been processed.

There is an important implication of this dual functionality. Because E-MOPs organize both the generalized information used for processing and items which are processed, both are equally accessible upon retrieval of a particular E-MOP. In addition, as new items are added to memory, new generalized knowledge is acquired. This integrated memory organization suggests the possibility of an understanding system fully integrated with long term memory. Because generalized and episodic knowledge are equally accessible, both can be used during the understanding process.

IPP (Lebowitz, 1980), which has been developed concurrently with CYRUS, is an integrated understanding system. It uses MOPs to hold generalized information about international terrorism and also to organize terrorism episodes it has read about. As it reads stories, it adds them to memory, creating new sub-MOPs and building up generalized information (similarly to the way CYRUS does). It uses the generalized information it builds up to make inferences while understanding subsequent stories. For example, after IPP has read a number of stories about kidnapping in Italy in which businessmen are kidnapped and held for ransom, it makes the generalization that "when there is a kidnapping in Italy, the victim is usually a businessman". Upon reading a later story about a kidnapping in Italy, if the victim's occupation is not mentioned, IPP will use that knowledge to make the inference that the victim was probably a businessman. In previous understanding systems which did not remember what they had read, and which did not update their generalized information, an inference such as the one above could

not have been made.

While the memory organization and generalization processes in IPP and CYRUS are similar, IPP's emphasis has been on using its generalized knowledge in later understanding, while CYRUS' has been on retrieval and reconstruction of episodes from memory.

9.4 Retrieval from conceptual memory

Because previous A. I. research into conceptual memory has centered on story understanding, approaches within A. I. to retrieval from conceptual memory have emphasized search of story representations or retrieval from specified conceptual structures, and not search of a large memory. These approaches have centered on question-answering, as opposed to memory search. A context for search, in those systems, is always immediately available, and questions are answered by using that context. Thus, the problem of searching memory initially for a relevant context has, for the most part, not been addressed. What has been addressed is the problem of how to interpret what kind of information a question is asking for, and what kinds of processing have to be done to extract that information from a known context. Some of the people who have addressed these QA problems have been Lehnert (1978), Scragg (1975), and Woods (1972).

Lehnert (1978) has defined 15 conceptual categories for questions. Each question category requires different processing for extraction of a good answer. "Causal antecedent" questions, for example, usually ask "why" and require backwards chaining through causal links in the appropriate event to find an answer. Thus, to answer "why did the the plane crash?", after a representation of a story has been formulated, the backwards causal connections from the event "the plane crashed" (in the story representation) would be searched. "Concept completion" questions, on the other hand, require only that a component (or slot filler) of the concept being asked about be retrieved from the memory representation of the appropriate event. Thus, to answer "where did the plane crash", the same planecrash event as above would be found in the story representation, and its place would be extracted from it.

Another problem addressed by Lehnert was the problem of inferential analysis of questions and giving the appropriate amount of information in an answer. When somebody asks a yes/no question, they normally want more than just a yes or a no. They are looking for additional explanation of the answer. Thus, a more appropriate answer than "no" to "Is Edmund Muskie a senator?" would be "No, he is secretary of state". A more appropriate answer than "yes" to "Is Muskie secretary of state?" would be "Yes, since the middle of May". By the same token, when somebody asks "Do you have the time?", they are not asking for a yes or no answer at all, but want to know what time it is. Lehnert's question answerer was implemented to answer questions about stories processed by the SAM program (Cullingford, 1977), which processed script-based stories (Schank and Abelson, 1977). I later extended it to answer questions about plan-based stories processed by PAM (Wilensky, 1978).

More recently, Dyer and Lehnert (1980) have been working on retrieval of information from more complicated stories than were dealt with in SAM and PAM. They have defined sets of access functions for accessing available memory representations and story components. Their starting point, like that of Lehnert's original question answering work, has been an already available story context for search. Thus, they have not looked into the problem of storing these story representations in long-term memory and retrieving them from among other similar contexts in memory. Though they have begun dealing with retrieval as a memory search problem, they are dealing with depth-wise search of a given context. The emphasis of this research has been on breadth-wise search for appropriate contexts in long-term memory.

Some other question-answering work which relates to that presented here is Collins' (1978) plausible reasoning research. He has defined 30 types of plausibility inferences people make in answering questions. Many of those inference types have been implemented in the SCHOLAR system (Carbonell and Collins, 1973). Collins, however, has not tried to integrate his inference types into a full process model of retrieval. He has not defined strategies for controlling the inferences, nor has he defined a memory organization or memory search processes which support them. Some of the inferences he defines fit into the reconstructive model in obvious places. His "lack of knowledge" inference, for example, is used by the traversal/elaboration process to decide that a particular fact is not in memory. Other default reasoning rules he defines fit into the reconstructive model as instantiation rules.

9.5 Organizing large amounts of knowledge in a computer memory

Within computer science, the fields of information retrieval and data base management have been concerned with organizing large amounts of information in the computer memory. Their concerns, however, are much different than those presented here. In this section, other approaches within computer science to memory organization and retrieval will be presented.

9.5.1 Conceptual memory and information retrieval

Although CYRUS has been presented as a model for an intelligent information retrieval system, it bears little resemblance to current information retrieval systems. Up to now, I. R. has been concerned with document retrieval and good key word schemes for classifying documents, and has not addressed the issues involved in retrieving facts. Because the emphasis has been on systems which could be used immediately, the techniques that have been developed for information retrieval have been statistically-based keyword schemes that could be applied independently of the particular domain which documents in the system belonged to.

Of course, some of the problems in storing large amounts of information are the same, whether key word schemes or conceptual categories are used. In particular, a major problem that has been addressed both here and in information retrieval has been the problem of how to retrieve items without enumerating large categories. In this thesis, the solution to that problem has involved both organizational and retrieval strategies. Organizational strategies are used during memory update to index and sub-divide memory categories. During retrieval, retrieval strategies elaborate on search keys and search for alternate contexts.

In information retrieval, techniques for non-enumeration have also been developed. As in this thesis, those techniques involve both category reorganization techniques during update and elaboration during retrieval. The category reorganization and elaboration used in IR are quite different than that presented here, however. In IR systems, categories are divided by applying statistical methods to the key words in the stored documents (Heaps, 1978). The groups of key words which divide a category into usefully-sized pieces are used. When a category is divided, each document is stored in only one of the new categories (Heaps, 1980). Elaboration in IR systems involves use of synonym lists, word truncation, universal symbols, and associated indices -- all key word schemes (Salton, 1975).

Although the techniques have been different, the goals of information retrieval are very close to the goals of the research presented in this thesis: to store and retrieve large amounts of relatively unconstrained information using a computer. Ultimately, this requires developing systems which can (1) understand the contents of documents and automatically organize them in the computer memory, and (2) retrieve the facts found in the documents when queried in Natural Language. Although this research does not resemble current information retrieval (Heaps, 1978), it is meant to provide a framework for future research in intelligent information retrieval (Schank, et al., 1980).

9.5.2 Data base management

Research in data base management has had two major aims: producing tools for storage and retrieval of tabular types of information; and developing mathematical models to prove that those systems are complete, sufficient, and secure. The relational model (Codd, 1979) lends itself to storage of tabular information and is the model most often used for those sorts of proofs.

One of the primary principles of data base management has been the decoupling of data base semantics from the content of a data base (Wiederhold, 1977). Semantics, in data base systems, refers to the meanings of the data base's relations, and their relationships to each other. Thus, a data base model, such as the relational model, provides a formalism for storage and manipulation of tabular information without having to specify exactly what the relations should be for any domain. The reason data base semantics and content have been decoupled is to insure that the same data base manipulations could be performed

regardless of the data base content.

Decoupling data base semantics from the data, however, makes it hard to query the data base about the kinds of data it holds. It also makes it hard to automatically re-organize the data on command. It became clear with the call for better interfaces to data bases that this separation was more of a detriment than an advantage. People designing natural language front ends for those data bases had to develop their own semantic models of the data base contents (Hemphill & Rhyne, 1978).

One data base researcher who has been interested in extending the semantics of the relational data base model so that it can be used for intelligent fact retrieval is Codd (1979). He has been attempting to define an extension to the relational model which is capable of capturing more of the meaning of data in the data base while preserving the independence of the implementation. In other words, he is interested in designing a system capable of more sophisticated storage and retrieval than has previously been the case with the relational model which will be applicable independent of the domain of information being stored.

In his extensions to relational data base model, Codd (1979) has proposed hierarchical relationships between memory categories and time relationships between events in a data base. Thus, he has begun to address the problem of episodic organization in a relational data base. Codd, however, does not go far enough in his data base extensions. Events, for example, can be related through causality and containment; his addition addresses only their time relationships. The additions Codd proposes to the relational algebra in order to support hierarchical organizations of categories support only strict hierarchies, and do not support specification of the relationship between the parent and child categories. Thus, a category can be specified as a generalization of another category without specifying which feature(s) of the child category it is generalized from. It also does not address the fact that some categories in a hierarchy might not inherit all properties from their parent categories. Although a category "meetings" might specify that the location of meetings is conference rooms, its sub-category "meetings with members of secret organizations" might specify that their locations are usually forests.

He does not address the problem of associating generalized information with categories directly in the data base, and he mentions as a problem but does not address the issue of automatic data base reorganization.

9.6 Summary

The theory of reconstructive memory that has been presented in this thesis is meant to be both a model of human retrieval processes and a model of intelligent information retrieval. As an information retrieval or data base system, however, it bears little resemblance to current systems. Information retrieval systems have based their memory organizations on keywords and not on conceptual categories. Because they do not organize their contents according to similarities in meaning, they cannot apply meaning-based heuristics for retrieval or category reorganization.

Data base management has been concerned more with data base architecture than with organization of episodic information. One exception to that is Codd's work, which begins to address some of the issues involved in episodic organization. He is attempting, however, to describe an architecture which will support organization of events in memory, as opposed to saying how particular events should be organized with respect to each other.

Within psychology, approaches to long term memory have described it as a reconstructive process. The processes that have been described by psychologists, however, have been described in very general terms independent of a memory organization or a description of the knowledge guiding the processes. This research, which explains a memory organization and particular retrieval processes, can be thought of as complementary to the psychological research of Norman and Bobrow, and Williams.

In A. I., previous research has been concerned with the processes and knowledge necessary for understanding, but not with integrating newly learned events into a long term memory for use in later understanding. This research, which attacks the problem of integrating new information into memory and later retrieving them, is the natural next step in Natural Language Understanding Research.

CHAPTER 10

Conclusions and Future Work

10.1 Conclusions

This thesis has presented a theory of reconstructive memory processing. The organization of such a memory, strategies for maintaining its organization, and strategies for retrieval have been presented. The memory organization described is one where items are are arranged in memory in conceptual categories, and differentiated within those categories according to their differences. Retrieval is a process of construction and elaboration of contexts for search. In this chapter, conclusions and possible extensions to the theory will be presented.

10.1.1 Non-enumeration

Perhaps the most important conclusion to be drawn from this research is the following:

(1) A few powerful strategies for search, combined with rich conceptual indexing, allow retrieval without category enumeration.

Two types of retrieval strategies have been presented in this thesis: instantiation strategies, which construct and elaborate contexts for search, and search strategies, which guide search for and within alternate contexts. In order to apply those strategies, the following types of knowledge are necessary:

1. context to context relationships

2. component to component relationships

3. relationships between types of components and contexts for search

In order to retrieve events from a category, memory's organization must provide discriminability between events. In a memory with that organization, specification of a unique set of features will allow an item to be retrieved. Thus, although retrieval does not involve category enumeration, a series of similar events can be found by reconstructing possible details, and searching for events with those details, or by searching for an alternate context that might refer to a target event.

The potential problem in a richly indexed system is that indexing takes up space and can get out of hand. In CYRUS, that problem was handled by the notions of salience, predictability, and generalization. Indices are created only for salient features of a context. After a number of events with the same feature are indexed, that feature is maintained as an index only if it proves to have predictive power. As events are indexed in sub-MOPs, relevant similarities between the events in those sub-MOPs are computed and stored on each sub-MOP as generalized information about the E-MOP. Features common to events in an E-MOP are not indexed.

In order to maintain retrievability as memory grows, new memory categories must be created, and generalized knowledge must be associated with them. Because new categories are created when multiple events are indexed in the same way, the same generalization process that controls index creation also constrains the creation of new categories. Generalized knowledge is built up by extracting the salient similarities between items in a sub-category. Newly-created generalized knowledge aids the retrieval process and constrains subsequent indexing within the sub-category.

10.1.2 Organizing generalized knowledge in memory

An intelligent system needs generalized knowledge, and it should be integrated into memory by associating it with the items it is generalized from. When a category is encountered during memory traversal, its indices and its generalized information are equally available. Additional search outside a data base for "meta-level" information would not be necessary to retrieve the generalized information needed for further processing. It would be available at the same level as the indices to the items themselves. In addition, associating generalized knowledge with items it is built up from allows it to be refined and updated as new examples are entered into memory. This organization of generalized knowledge has an important processing implication:

(2) Associating generalized information with the items it is generalized from makes retrieval of generalized information the same as retrieval of actual items.

In the model described, memory categories organize similar items, and also have generalized information associated with them describing the similarities between those items. Thus, _generalized knowledge is organized in the same hierarchical, inter-connected structures which organize the items themselves_. Retrieval of a category requires the same reconstructive processing as retrieval of an event from a category. In terms of the human model, this supports the theory that _there is no distinction between episodic and semantic memory_ (Schank, 1975). Both types of information are integrated into the same memory structures, and retrieval of both is the same.

It also has implications in the design of computer systems. In a computer system which makes retrieval of generalized information the same process as retrieval of actual items, a user can retrieve information about the nature of the data base, and not only data items themselves. Using the retrieval strategies for retrieving actual items from memory, and using the CYRUS data base, answering questions such as the following would be a trivial extension to the system.

(Q10-1) When Vance goes on diplomatic trips, where does he usually stay?

(Q10-2) Do welcoming ceremonies for Vance usually include parades?

(Q10-3) What kinds of things does Vance discuss at diplomatic meetings?

Instead of retrieving actual events that correspond to the target event in each of these cases, E-MOPs which correspond will be retrieved, and their generalized information will be consulted.

In an information retrieval system which did not allow queries about the nature of the data, a user would have to request all or many instances of the kinds of events he was interested in and draw his own conclusions. Retrieving individual items would require elaboration by the system or a series of questions by the user. Either way, it would be less efficient.

10.1.3 Retrieval failures and ease of retrieval

The theory presented makes the following psychologically relevant prediction:

(3) The closer a retrieval cue is to encoding at time of retrieval, the easier retrieval should be.

An event can be retrieved only by specifying features on which it was discriminated during processing. When an event specification refers to features that have not been discriminated, retrieval can only occur if the specified features can be transformed into features that have been discriminated. Psychologists have called this phenomenon _encoding specificity_ (Tulving, 1972). According to that theory, the ease with which an item can be recalled from memory depends on the nearness of the retrieval specification to the description initially encoded in memory for the item.

It must be taken into account, however, that memory is constantly reorganizing itself over time. Thus, it is the indices for an event at the time of retrieval, and not at time of input which determine how easy it is to retrieve it. Any time a retrieval specification refers to features of events which have not been discriminated, it will be necessary to elaborate or reconstruct aspects of the retrieval specification to correspond to features that are discriminated the conceptual category.

Ease of retrieval depends on ease of finding unique indices: retrieval failure occurs if necessary cues are not available for construction of an appropriate retrieval context. Because reconstructive retrieval processes are knowledge-based, if the knowledge necessary to construct or elaborate a context appropriately is not available at the time of reconstruction, then retrieval will fail.

10.2 Extensions and future work

There are a number of extensions that can be made to the theory presented, both in terms of a theory of memory organization and in terms of new domains and applications. In this section, those extensions will be discussed.

10.2.1 Extensions to a theory of long term memory

In the memory model which has been presented, events are retrieved reconstructively. Little has been said, however, about the control of retrieval strategies. In a fully reconstructive memory, not only should retrieval of events be reconstructive, but control and retrieval of the strategies themselves should also be reconstructive. Choosing the kinds of indices to elaborate during retrieval, for example, may be guided by choosing a prototypical event from a memory category and choosing from among its salient features.

In the theory described, memory reorganization happens as new events are added to memory. A possible extension to that is reorganization because of retrieval access. If a particular kind of information is requested often, for example, memory should reorganize itself to make that information more available. Recently accessed and rehearsed information is easier for people to retrieve than unrehearsed information (Bjork & Whitten, 1974, Crowder, 1976). What this suggests is that each time memory is accessed, whether for retrieval or update, it is reorganized taking recency information into account.

In CYRUS, this has been partially realized through maintenance of a dialog context, and by keeping track of the most recent event added to an E-MOP. Information available from the context of a previous answer is easier to retrieve than other information. A most recent experience is easier to retrieve than other experiences like it. Recency has not been addressed in the general case, however. When an event is no longer the most recent of its type, or no longer in the immediate conversational context, its recency has no effect on its ease of retrieval.

Another extension which could be made to the theory presented is extensive development of organizations and strategies relating to time. As in indexing other aspects of events in E-MOPs, it is the predictive features of time specifications which should be used as indices. Besides indexing events in E-MOPs, they can be organized according to time. The "era" was developed for CYRUS to organize events by time, but has not been extended fully. An era breaks a time line into reasonably-sized chunks, providing conceptual categories for storing generalized information about a time period. Strategies which would use time for retrieval would use the generalized information associated with eras, the relationships of eras to each other, as well as the placement of landmark events in eras (Kolodner, 1978).

10.2.2 Information retrieval

The theory presented has been referred to as the basis for intelligent information retrieval. CYRUS, as an expert on Cyrus Vance, is the first step in designing an expert on current events. In order to make CYRUS into a current events expert, there are a number of additions that would have to be made. First, its domains of expertise would have to be expanded. Second, it would have to be able to access multiple data bases about numerous people. Third, its organizational structures would have to be extended to handle information about countries and political entities, in addition to handling information about people. It would also need to have a better notion of time than has been implemented in CYRUS.

Although this might seem like a lot of additions to be made to the implementation, the theory behind it would have to be changed very little. The retrieval and organizational strategies presented here would form the core of its memory processing. The same notions of salience, predictibility, and generalization which are necessary to differentiate day to day events are necessary to differentiate world events. The major addition to the theory would have to be the development of time organizations and strategies, which have not been discussed extensively here.

10.2.3 Expert systems

Another possible application for an episodic memory is in the design of expert systems. Research in expert systems has focused on rule application (Aikens, 1980, Davis, et al., 1977). Suppose, however, that we wanted to design an expert system based on experiential information. In an experiential system, experiences would be retrieved from memory to find a course of action, rather than retrieving explicit rules. If a course of action had worked in one situation, it could be found again and used in a similar situation.

The episodic memory organization and strategies for search presented in this thesis could provide a basis for such a system. Organizational structures in the system would correspond to situations from the expert domain. In a medical system, for example, "treating hepatitis" might be one of the organizational structures. That structure would store a description of the normal symptoms of the desease and a procedure for treating the general case. Specializations of "treating hepatitis", such as "treating mild hepatitis" or "treating hepatitis in diabetics", would be indexed off of "treating hepatitis" by their differences from its norms. If a case of hepatitis matched one of those specializations, the treatment associated with that specialization would be suggested.

In particular, such a system would handle exceptions to rules in a very nice way. The general case would be stored at the category level, with exceptions indexed off of it. In a medical system, in any but the exceptional cases, the general treatment would be retrieved. If a new case referenced a previous exception, knowledge about the previous

exception would be used to suggest a treatment. When a treatment was successful a number of times for a particular exception, it would be generalized as the treatment for that exception. The general rule would not have to change, unless an exception became the generalized case.

Just as generalized information about episodes changes with experience, the norms for recognition and treatment of a desease would have to change with experience. Applying the rules for memory reorganization presented in chapter 7 would allow that to happen.

The medical domain, and other expert domains which require a decision to be made, are different from the domain presented in this thesis in one important way. Cause and effect are very important in expert domains. A disease must be recognized and then treated. If the treatment doesn´t work, it must be determined why not, and generalized information must be updated. Thus, in addition to the strategies and organization presented in this thesis, failure-based memory reorganization (Schank, 1981) and learning processes, as well as a theory of explanation, would have to be developed.

APPENDIX A

Context-to-Context Instantiation Strategies

These strategies compute contexts related to a specified event. Their first step is to retrieve the appropriate knowledge from the content frame of the E-MOPs the input event fits into. They infer as much as possible from the specified relationships between the E-MOP the input event belongs to and that of the event to be output. After applying domain-specific knowledge, generalized information about the event-relationship between the two events is used to fill in details. Event relationships include containment, causality, and time relationships. In the last step, component instantiation strategies are called to fill in more details.

The complete list of context-to-context instantiation strategies, each of which corresponds to a content frame specification (presented in chapter 5), is listed below (repeated from chapter 3):

Context-Instantiation Strategies

Instantiate-Enablements
Instantiate-Preconditions
Instantiate-Results
Instantiate-Reasons
Instantiate-Enabled-Events
Instantiate-Larger-Episodes
Instantiate-Seq-of-Events
Instantiate-Preceding-Events
Instantiate-Following-Events
Instantiate-Standardizations

Figure A-1

A.1 Containment and contained-in relationships

Instantiate-Seq-of-Events, which is used to construct specifications of events that might be included in an event, is repeated from chapter 3 below:

Instantiate-Seq-of-Events

(1) Identify the E-MOP the specified event fits into.

(2) Get the default sequence of events for that E-MOP.

(3) Go through the event sequence one by one transferring components as specified and making the following inferences when more specific information is not available on the E-MOP.

 (a) Time specifications are within the time specified on the input event.

 (b) Place specifications are within the place specified on the input event.

 (c) Participants in any of the events being instantiated include a subset of the known participants plus a number of as yet unspecified participants.

(4) Use relevant Component-Instantiation strategies to further specify components of each event, using the already-instantiated partial descriptions of each component as constraints.

Figure A-2

The strategy "Instantiate-Larger-Episodes" constructs contexts for larger episodes an event could be part of. It uses specifications on E-MOPs the target episode fits into to choose contexts for larger episodes the target could be part of. Notice that the major difference between the strategy and the one presented above is in step 3. In the strategy above, knowledge about how larger episodes relate to the episodes they include was used. In the strategy below, general knowledge about how smaller episodes fit into larger ones is used to adjust the time, place, and participants of the episode. The time of a larger episode includes the time of the smaller one. Its place includes the place of the smaller one. Its topic includes that of the smaller.

Instantiate-Larger-Episodes

(1) Identify the E-MOP the specified event fits into.

(2) Get the default larger episodes for that E-MOP.

(3) For each of those specifications, transfer components as specified and make the following inferences when more specific information is not available on the E-MOP.

(a) Time specifications include the time specified on the input event.

(b) Place specifications include the place specified on the input event.

(c) Participants of the input event are included in the participants of the larger event, groups and organizations they belong to might also be involved.

(d) Topic of the larger episodes includes that of the input.

(4) Use relevant Component-Instantiation strategies to further specify components of each event, using the already-instantiated partial descriptions of each component as constraints.

Figure A-3

Imagine, for example, attempting to retrieve sightseeing experiences in the Middle East. In order to do that, it might be appropriate to recall trips to the Middle East and sightseeing during those trips. Trips are *larger episodes* the sightseeing could be part of. In order to retrieve those trips, contexts for their retrieval must first be constructed. In constructing those contexts, contextual components must be transferred from the target concept (in this case sightseeing) to the new retrieval concept (in this case a trip). In this example, the place of the target sightseeing episode must be transferred to the trip instantiation. The actor of the sightseeing should be the same as the actor of the trip. Some of the other participants in the target sightseeing episode might be participants in the trip episode. And if the time of the sightseeing episode is known, the time of the trip can be inferred from it. After construction of the trip context, its retrieval can be attempted.

A.2 Preceding events

Enabling events, preconditions, and preceding events all happen before the target event they are being constructed from. Since those event relationships do not include containment, the relationships between other aspects of the input event and event to be constructed do not have any generally defined relation. The strategies Instantiate-Enablements, Instantiate-Preconditions, and Instantiate-Preceding-Events all have the same relationship in time to their target event -- they come before it. Step 3 of those strategies is listed below.

Instantiating Preceding Events

(3) For each of those specifications, transfer components as specified and make the following inferences when more specific information is not available on the E-MOP. The time of the new event is before that of the input event.

Figure A-4

Consider, for example, a person who has lost his glasses trying to recall where he could have left them. He might try to reconstruct his previous sequence of events until he recalls where his glasses might be. If he is at home while trying to recall where his glasses are, it would be appropriate to apply the strategy Instantiate-Enablements to construct the context for enabling his getting home. If he usually gets home by driving, he will construct a context for driving home and attempt to recall today's particular driving home episode and whether he had had his glasses or put them down somewhere during the drive. If that doesn't help him, he may apply the strategy Instantiate-Preceding-Events to construct a context for what he might have been doing before driving home. In applying each of these strategies, he must know that the event being constructed happened before the event he is constructing it from. That time specification can be used to help him infer times for the events being reconstructed.

A.3 Following events

In constructing resulting events, events a target event might enable, and events following a target, the time relationship of those events to the target is a "later than" relationship. Step 3 for Instantiate-Results, Instantiate-Enabled-Events (which infers events the target could enable), and Instantiate-Following-Events specifies that the default time of instantiated episodes is after the time of the target event.

Instantiating Following Events

(3) For each of those specifications, transfer components as specified and make the following inferences when more specific information is not available on the E-MOP. The time of the new event is after that of the input event.

Figure A-5

Suppose that instantiation strategies were used to recall ends of negotiations. The kinds of events that could result from ends of negotiations, and events that negotiations endings could enable would have to be constructed and searched for. Those include treaty signings and perhaps the beginnings of cold wars, results of aborted negotiations. To construct the contexts for those episodes, the strategies Instantiate-Results and Instantiate-Enabled-Events would be applied. Knowledge on relevant negotiations E-MOPs would help in instantiating possible treaty signings and cold wars. General information about the relationship of resulting events would allow inference of sketchy time specifications for those events -- sometime after the target negotiations episode.

A.4 Standard methods of doing an activity

Standardizations are standard methods of achieving an E-MOP's state. Standard types of meetings for Vance include diplomatic meetings, consultations, and public relations meetings. Standard types of diplomatic ceremonies include welcomes, parades, and treaty signings. Travel can be by plane, train, bus, etc. The time of a standardization is that of the specification event it is being constructed from. A standard event is merely a specialization of a more general E-MOP. Thus, its participants, place, and duration are also the same. In instantiating a standard way of doing an activity, those features are transferred directly from the target event. Instantiate-Standardizations is the instantiation strategy which guides construction of standard ways of doing an activity.

Instantiate-Standardizations

(3) For each of those specifications, transfer components as specified and make the following inferences when more specific information is not available on the E-MOP.

(a) Time specifications are the same as those of the input event.

(b) Place specifications are the same as those of the input event.

(c) Participants are the same as those of the input event, though there might be others in addition.

Figure A-6

In constructing a treaty signing ceremony from the specification that there was a ceremony at the White House including Begin, Sadat, and Carter, the *instantiated* treaty signing ceremony will have participants Begin, Sadat, and Carter and location the White House.

APPENDIX B

CYRUS' E-MOPs

CYRUS' E-MOPs correlate with the kinds of activities a secretary of state does. Below is a complete list of CYRUS' E-MOPs (also presented in chapter 5):

CYRUS' E-MOPs

scripts:
- $MEET -- diplomatic meetings with foreign dignitaries
- $CONSULT -- consultations with American officials
- $PUB-REL-MEET -- political meetings with non-diplomats
- $STATE-DINNER -- state dinners
- $FLY -- flights
- $REPORT -- press conferences
- $SPEECH -- speeches
- $PARTY -- parties
- $DINE -- dinner with *associates*

simple MOPs:
- sM-MEETING -- political meetings
- sM-CONFERENCE -- political conferences
- sM-SUMMIT-CONFERENCE -- specialized political conferences
- sM-TRAVEL -- traveling
- sM-TRIP -- generalized trip
- sM-VIPVISIT -- diplomatic trip
- sM-SIGHTSEE -- sightseeing
- sM-CEREMONY -- ceremonies
- sM-SOCIAL-POL-ACTIVITY -- social political activities
- sM-DIPLOMATIC-ACTIVITY -- diplomatic activities

IMOPs:
- I-NEGOTIATE -- negotiations
- I-DIPLOMACY -- diplomacy
- I-SOCIALIZE -- socializing
- I-CELEBRATE -- celebration

Figure B-1

CYRUS' E-MOPs fall into three MOP classes -- scripts, simple MOPs, and IMOPs. Scripts have a highly *stereotypical sequence of events*. Simple MOPs are less stereotypical. IMOPs (Intentional MOPs) have a

less stereotypical sequence of events, but their goals are stereotypical. For a better description of the distinction between these three categories, see section 4.3.3 of the thesis.

Within CYRUS, the information associated with E-MOPs is represented using Conceptual Dependency (CD) notation (Schank, 1975). CYRUS' representations of some of the content frame properties for $MEET ("diplomatic meetings") and sM-VIPVISIT ("diplomatic trips") are presented below:

$MEET

```
MOP-type: script
FRAME:
  ((<=> ($MEET ACTOR &ACTOR OTHERS &OTHERS TOPIC &TOPIC))
   PLACE &PLACE TIME &TIME DURATION &DUR))
&OTHERS: (CLASS (#PERSON) OCCUPATION (RT-DIPLOMAT))
&TOPIC:  (CLASS (#CONTRACT #EVENT))
&DUR:  (*ORDERHOURS*)
MOP-categories:  (sM-MEETING)
SMOPs:
  ((<=> (sM-CONFERENCE ACTOR &ACTOR OTHERS &OTHERS
                       TOPIC &TOPIC))
   PLACE &PLACE TIME &TIME)
  ((<=> (sM-VIPVISIT ACTOR &ACTOR DESTINATION &PLACE))
   TIME &TIME)
IMOPs:
  ((<=> (I-NEGOTIATE ACTOR &ACTOR OTHERS &OTHERS
                                  TOPIC &TOPIC)))
MAINCON:
  ((ACTOR &ACTOR <=> (*MTRANS*)
                 MOBJECT (*CONCEPTS* CONCERNING &TOPIC)
                 FROM &ACTOR TO &OTHERS)
   PLACE &PLACE MANNER (*BILATERAL*))
MOST-RECENT:
  ((<=> ($MEET ACTOR HUM1 OTHERS HUM62 TOPIC CNTRCT1))
   TIME TIM342 PLACE POL6)
```

Figure B-2

The FRAME property on any E-MOP specifies its slots and names variables for those slots. The atoms beginning with "&" are variables. The atoms before those are the names of the slots. Thus, $MEET has slots for ACTOR (the person from whose point of view the episode is stored), OTHERS (other participants), TOPIC, PLACE, TIME, and DURATION. Together, the ACTOR and OTHERS make up the PARTICIPANTS. Default values for each of those slots can be found as content frame properties on an E-MOP named by the name of the variable. The specification on $MEET for &OTHERS is that they are persons who are diplomats. The topic (&TOPIC) is either a contract or event, and the default duration (&DUR) is a

small number of hours (*ORDERHOURS*).

"MOP-categories" are more general categories the E-MOP falls into. $MEET is a "political meeting" (sM-MEETING), and thus inherits the properties of "political meeting". Because "political meeting" is both an occupational and political activity, $MEET does not have to specify that it falls into those categories. Those categories were described in chapter 6.

The "SMOPs" and "IMOPs" properties specify larger episodes diplomatic meetings are normally part of. These two properties are used during retrieval to direct memory search. The variable names in each of those specifications denote the correspondence between the slot fillers of those related episodes and those of a typical diplomatic meeting. As in Cullingford's (1977) scripts, the "MAINCON" gives the main activity in the sequence of events of the script. The main activity of $MEET is bilateral communication of information between the participants.*

In addition, each E-MOP in CYRUS keeps track of the last episode that was added to the E-MOP (specified by MOST-RECENT in the E-MOP above). Thus, most-recent events are treated specially and are more accessible than other events in the E-MOP. In this way, CYRUS can always retrieve at least one event from an E-MOP even if appropriate indices cannot be proposed by elaboration processes.

Information in the E-MOP sM-VIPVISIT ("diplomatic trips") is represented as follows:

```
                              sM-VIPVISIT

MOP-type:  simple MOP

FRAME:
 ((<=> (sM-VIPVISIT ACTOR &ACTOR DESTINATION &DEST
                    ORIGIN &ORIG OTHERS &OTHERS))
  TIME &TIME PLACE &DEST DURATION &DUR)

&OTHERS: (CLASS (#PERSON) OCCUPATION (RT-DIPLOMAT)
          NATIONALITY POL1)   (i.e., American diplomats)
&DEST: (CLASS (#LOCALE #POLITY))
&ORIG:  POL2          (i.e., Washington)
&DUR:  (*ORDERDAYS*)

MOP-categories:  (sM-TRIP)
```

---

*MTRANS is the CD primitive for "mental transfer of information". For more information about other CD primitives, see Schank (1975).

```
IMOPs:
  ((<=> (I-NEGOTIATE ACTOR &ACTOR OTHERS &NEG-OTHERS
                                  TOPIC &TOPIC)))
SEQ-OF-EVENTS:
  ((<=> ($FLY ACTOR &ACTOR DESTINATION &DEST ORIGIN &ORIG
              PARTY &OTHERS)))
  ((<=> ($WELCOME ACTOR &WELCOMER GUEST &ACTOR))
   PLACE &WEL-PLACE)
  ((<=> (sM-DIP-ACTIVITY ACTOR &ACTOR OTHERS &PARTY))
   PLACE &DEST)
          repeat: indefinitely
  ((<=> ($FLY ACTOR &ACTOR DESTINATION &ORIG ORIGIN &DEST
              PARTY &OTHERS)))
MAINCON:
  ((<=> (sM-DIP-ACTIVITY ACTOR &ACTOR OTHERS &PARTY))
   PLACE &DEST)

&NEG-OTHERS:  (CLASS (#PERSON) NATIONALITY &DEST
                OCCUPATION (RT-DIPLOMAT))
&TOPIC:  (CLASS (#CONTRACT) SIDES (DEST))
&WELCOMER:  (CLASS (#PERSON) NATIONALITY &DEST
              OCCUPATION (RT-DIPLOMAT))
&WEL-PLACE (CLASS (#LOCALE) TYPE (*AIRPORT)
             LOCATION &DEST)
&PARTY:  (CLASS (#GROUP)
           MEMBERS (&ACTOR &OTHERS &NEG-OTHERS))

MOST-RECENT:
 ((<=> (sM-VIPVISIT ACTOR HUM1 DESTINATION POL5
                    ORIGIN POL2))
  TIME TIM547 PLACE POL5)
```

Figure B-3

The FRAME, MAINCON, MOP-categories, IMOPs, and MOST-RECENT properties hold the same information explained above for $MEET. In addition, the SEQ-OF-EVENTS feature holds the normal sequence of events for a diplomatic trip. A trip includes flying to the destination, being welcomed at the airport, repeatedly doing diplomatic activities, and then flying home. The role filler specifications at the bottom of the diagram (after the MAINCON) are specifications of the role fillers for episodes related to trips. &WEL-PLACE, for example, is used in the $WELCOME scene of a sM-VIPVISIT, and holds the information that the place of a welcome is the airport of the country being visited.

References

Aikens, J. S. (1980). Prototypes and production rules: A knowledge representation for computer consultations. Ph D. thesis. Technical Report. Department of Computer Science. Stanford University, Palo Alto, CA.

Anderson, J. R. (1974). Retrieval of propositional information from a long term memory. Cognitive Psychology, Vol. 5, pp. 451-474.

Anderson, J. R. (1976). Language, memory, and thought. Erlbaum, Hillsdale, N. J.

Anderson, J. R. and Bower, G. (1973). Human Associative Memory. Winston-Wiley, Washington, D. C.

Bartlett, R. (1932). Remembering: A Study in Experimental and Social Psychology. Cambridge University Press, London.

Birnbaum, L. and Selfridge, M. (1979). Problems in conceptual analysis of natural language. Research Report #168. Department of Computer Science. Yale University, New Haven, CT.

Bjork, R. A., and Whitten, W. B. (1974). Recency-sensitive retrieval processes in long-term free recall. Cognitive Psychology, Vol. 6, pp. 173-189.

Bobrow, D. G. and Norman, D. A. (1975). Some principles of memory schemata. In Bobrow, D. G. and Collins, A., ed. Representation and Understanding. Academic Press, Inc., New York.

Borko, H. and Bernier, C. L. (1978). Indexing Concepts and Methods. Academic Press, Inc., New York.

Bower, G. H., Black, J. B., and Turner, T. J. (1979). Scripts in Text Comprehension and Memory, Cognitive Psychology, Vol. 1, 177-220.

Carbonell, J. R. and Collins, A. (1973). Natural semantics in artificial intelligence. Proceedings of Third International Joint Conference on Artificial Intelligence, 1973, pp. 344-351.

Charniak, E. (1977) A framed painting: the representation of a common sense knowledge fragment. Cognitive Science, Vol. 1, pp. 355-394.

Codd, E. F. (1979). Extending the database relational model to capture more meaning. Paper presented at 1979 ACM SIGMOD Conference, Boston, Mass.

Collins, A. (1978). Studies of plausible reasoning, final report. Volume I: Human plausible reasoning. BBN Report 3810, Bolt Beranek and Newman, Inc., Cambridge, MA.

Collins, A., Warnock, E. H., Aiello, N., and Miller, M. L. (1975). Reasoning from incomplete knowledge. In Bobrow, D. G. and Collins, A., ed. Representation and Understanding. Academic Press, Inc., New York.

Crowder, R. G. (1976). Principles of Learning and Memory. Lawrence Erlbaum Associates, Hillsdale, New Jersey.

Cullingford, R. (1978). Script application: computer understanding of newspaper stories. Ph. D. thesis. Research Report #116. Department of Computer Science. Yale University, New Haven, CT.

Davis, R., Buchanan, B. G., and Shortliffe, E. H. (1977). Production rules as a representation for a knowledge-based consultation program. Artificial Intelligence. Vol. 8, pp. 15-45.

DeJong, G. F. (1979a). Skimming stories in real time: An experiment in integrated understanding. Research Report #158. Department of Computer Science. Yale University, New Haven, CT.

DeJong, G. F. (1979b). Prediction and substantiation: a new approach to natural lnguage processing. Cognitive Science, Vol. 3, pp. 251-273.

Dyer, M. G. and Lehnert, W. G. (1980). Organization and search processes for narratives. Research Report #175. Department of Computer Science. Yale University, New Haven, CT.

Gibbs, R. W. and Tenney, Y. J. (1980). The concept of scripts in understanding stories. Journal of Psycholinguistic Research. Vol. 9, pp. 275-284.

Heaps, H. S. (1978). Information Retrieval: Computational and Theoretical Aspects. Academic Press, New York.

Hemphill, L. and Rhyne, J. (1978). A Model for Knowledge Representation in Natural Language Query Systems, IBM Research Report RJ2304(31046).

Kolodner, J. L. (1978). Memory organization for natural language database inquiry. Research Report #142. Department of Computer Science. Yale University, New Haven, CT.

Lebowitz, M. (1980) Generalization and Memory in an Integrated Understanding System. Ph. D. Thesis. Research Report #186. Department of Computer Science. Yale University, New Haven, CT.

McGuire, R. (1980). Political primaries and words of pain. Unpublished manuscript. Department of Computer Science. Yale University, New Haven, CT.

Newell, A. (1973). Artificial intelligence and the concept of mind. In Schank. R. C. and Colby,K. M . Computer Models of Thought and Language. Freeman and Company, San Francisco, CA.

Norman, D. (1972). Memory, knowledge, and the answering of questions. Center for Human Information Processing Memo CHIP-25. University of California at San Diego.

Norman, D. A. and Bobrow, D. G. (1977). Descriptions: a basis for memory acquisition and retrieval. Report #7703. Center for Human Information Processing, La Jolla, California.

Owens, J.,Bower,G. H., and Black, J. B. (1979). The "soap opera" effect in story recall. Memory and Cognition, Vol. 7, pp. 185-191.

Quillian, M. R. (1968). Semantic memory. In M. Minsky , ed. Semantic Information Processing.MIT Press, Cambridge, MA.

Reder, L. M. and Anderson, J. R. (1980). A partial resolution of the paradox of interference: the role of integrating knowledge. Cognitive Psychology, Vol. 12, pp. 447-472.

Salton, G. (1975). Dynamic Information and Library Processing. Prentice-Hall, Englewood Cliffs, New Jersey.

Schank, R. C. (1974). Conceptual Information Processing. North Holland, Amsterdam.

Schank, R. C. (1975). The structure of episodes in memory. In Bobrow, D. G. and Collins, A., ed. Representation and Understanding. Academic Press, Inc., New York.

Schank, R. C. (1979). Reminding and memory organization: An introduction to MOPs. Research Report #170. Department of Computer Science, Yale University.

Schank, R. C.(1981). Failure-driven memory. Cognition and Brain Theory. Vol. 1.

Schank, R. C. and Abelson, R. P. (1977). Scripts, Plans, Goals, and Understanding. Lawrence Erlbaum Press, Hillsdale, N.J.

Schank, R. C. and Kolodner, J. L. (1979). Retrieving Information from an Episodic Memory Research Report #159. Department of Computer Science, Yale University. Short version in Proceedings of the Sixth International Joint Conference on Artificial Intelligence, Tokyo.

Schank, R. C. , Kolodner, J. L. , and DeJong, G. (1979). Conceptual Information Retrieval. In Proceedings of Research and Development in Information Retrieval Conference, .St. John's College, Cambridge, England.

Smith, E. E., Adams, N., and Schorr, D. (1978). Fact Retrieval and the Paradox of Interference, Cognitive Psychology, 10, pp.438-464.

Spiro, R. J. (1979). Prior knowledge and story processing: integration, selection, and variation. Techinical Report #138. Center for the Study of Reading. Illinois.

Tulving, E. (1972). Episodic and semantic memory. In E. Tulving & W. Donaldson, ed. Organization of Memory. Academic Press, New York.

Weiderhold, G. (1977). Database Design. McGraw-Hill Book Company, New York.

Wilensky,R. (1978). Understanding goal-based stories. Ph. D. Thesis. Research Report #140. Department of Computer Science. Yale University, New Haven, CT.

Williams, M. D. (1978). The process of retrieval from very long term memory. Center for Human Information Processing Memo CHIP-75, La Jolla, California.

Woods, W., Kaplan, R., and Nash-Webber, B. (1972). The LUNAR sciences natural language information system: final report. Bolt, Beranek and Newman Report No. #2378, Cambridge, MA.